MANGALORE UNIVERSITY

Final Synopsis

of the PhD Thesis to be submitted for the award of degree of
Doctor of Philosophy in

Computer Science

TITLE OF THE WORK

Improved Data Analysis for E-Commerce Customers using Sentiment Analysis

By

Amrithkala M Shetty
Research Scholar
Registration No: MU/EXB/Ph.D/CR.33/Comp.Sc./2022-23/E.25 dated 14.06.2023

Under the Guidance of
Dr. Manjaiah D. H.
Senior Professor

Department of Post-Graduate Studies and Research in Computer Science,
Mangalore University,
Mangalagangotri-574 199

September 2025

Contents

1 Hypothesis and Research Issue **2**

1.1 Introduction . . . 3

1.2 Sentiment analysis (SA) Approaches . . . 4

1.2.1 Machine learning approaches . . . 4

1.2.2 Deep learning approaches . . . 4

1.2.3 Hybrid approaches . . . 4

1.2.4 Transformer based approaches . . . 5

1.3 Challenges in Sentiment Analysis Across Approaches . . . 5

1.4 Literature Review . . . 6

1.4.1 Sentiment analysis using Lexicon based techniques . . . 6

1.4.2 Sentiment analysis using Machine learning techniques . . . 6

1.4.3 Sentiment analysis using Deep Learning techniques . . . 8

1.4.4 Sentiment analysis usinf Transformer Based techniques . . . 9

2 Objectives **10**

3 Proposed Methods **10**

3.1 Hyperparameter Optimization of Machine Learning Models using grid search for Amazon Review Sentiment Analysis . . . 10

3.1.1 Proposed methodology . . . 11

3.1.2 Experiments . . . 12

3.1.2.1 Experimental setup . . . 12

3.1.3 Results analysis . . . 12

3.2 Unleashing the Power of 2D CNN with Attention and Pre-trained Embeddings for Enhanced Online Review Analysis . . . 20

3.2.1 Methodology . . . 20

3.2.2 Proposed model . . . 20

3.2.2.1 Attention layer . . . 21

3.2.3 Dataset . . . 22

3.2.4 Data annotation . . . 23

3.2.5 Evaluation metrics . . . 23

3.2.6 Result analysis . . . 24

3.2.6.1 Proposed Model Performance on Balanced and Unbalanced ACR Datasets . . . 25

3.2.6.2 Proposed Model Performance on Balanced and Unbalanced AKR Datasets . . . 25
3.3 Fine-Tuning XLNet for Amazon Review Sentiment Analysis: A Comparative Evaluation of Transformer Models . . . 29
3.3.1 Framework Methodology . . . 32
3.3.1.1 Fine-tuned XLNet model . . . 32
3.3.2 Result analysis . . . 34
3.3.2.1 Performance comparison of DL models on AE dataset . . . 34
3.3.2.2 Comparative analysis of transfer learning models on the AE dataset . . . 35
3.3.2.3 Performance comparison of DL models on the AP dataset 37
3.3.2.4 Comparative analysis of transfer learning models on the AP dataset . . . 37
3.4 Enhancing Sentiment Analysis of Amazon Reviews with Deep Hybrid Parallel Fusion of CNN and LSTM Using Pretrained Word Embeddings . . . 39
3.4.1 Proposed model (HPFCL) . . . 40
3.4.1.1 CNN-based feature extraction . . . 41
3.4.1.2 LSTM-based feature extraction . . . 41
3.4.1.3 Hybrid Fusion of CNN+LSTM . . . 42
3.4.1.4 Classification . . . 42
3.4.2 Results and analysis . . . 43
3.4.2.1 Experimental analysis on GloVe . . . 43
3.4.2.2 Experimental analysis on FastText . . . 43
3.4.2.3 Experimental analysis on Word2Vec . . . 44

4 Conclusion and Future works **46**

5 Publications **47**

Improved Data Analysis for E-Commerce Customers using Sentiment Analysis

Abstract

In the rapidly evolving landscape of e-commerce, understanding customer sentiment is crucial for enhancing user experience and driving business decisions. This thesis focuses on advancing sentiment analysis (SA) techniques to improve data analysis for e-commerce platforms. The primary objective is to develop and evaluate effective SA models that can accurately classify user opinions and extract valuable insights from product reviews. To achieve this, we address several key research objectives: (1) developing a comprehensive sentiment analysis framework tailored for e-commerce platforms, (2) classifying user opinions into distinct sentiment categories using machine learning (ML) algorithms, (3) investigating the application of deep learning (DL) approaches for sentiment analysis on e-commerce datasets, and (4) fine-tuning the XLNet model for Amazon review sentiment analysis and comparing its performance against other transformer-based models.

We propose a novel approach leveraging a Two-Dimensional Convolutional Neural Network with an Attention mechanism (Att+2D CNN) to analyze Amazon cell phone and Kindle reviews. This model, enhanced with pre-trained word embeddings (GloVe and Word2Vec), is designed to capture intricate patterns in text data. Traditional one-dimensional convolutional neural networks (1D CNNs) are effective in capturing local features between words, but they may overlook broader contextual relationships. By extending this to two dimensions and incorporating an attention mechanism, our Att+2D CNN model can better capture patterns across different dimensions of text, ensuring that critical information is retained even when processing large datasets. We compare its performance against various DL models, including 1D CNN, 2D CNN, LSTM, BiLSTM, GRU, and BiLSTM with attention, to demonstrate its efficacy on both balanced and unbalanced datasets. Our results show that the Att+2D CNN model, combined with pre-trained embeddings, achieves superior performance in sentiment classification tasks.

In addition to DL approaches, we explore hyperparameter optimization using grid search to enhance the performance of six ML models (Decision Tree, Logistic Regression, Naive Bayes, Support Vector Machine, XGBoost, and Random Forest) for Amazon review SA. Hyperparameter tuning is a critical step in building accurate and efficient ML models, as it directly impacts their ability to generalize to unseen data. By employing feature extraction techniques such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Average Word2Vec and Word2Vec, we optimize model hyperparameters to improve accuracy, precision, recall, and F1-score. Our findings highlight the significant impact of hyperparameter tuning on model performance, with optimized models consistently outperforming their baseline counterparts. For instance, the combination of TF-IDF with Word2Vec as a feature extraction method, coupled with grid search

optimization, yielded the highest accuracy for several ML models. We also evaluate model performance using metrics such as AUC and confusion matrices, providing a comprehensive assessment of their effectiveness.

We propose a novel two-layered sentiment classification framework that introduces a Hybrid Parallel Fused CNN-LSTM (HPFCL) model as the base learner. This deep hybrid architecture integrates Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks in parallel, enabling the model to capture both local word-level features and long-range dependencies. The outputs from the parallel branches are fused into a unified representation, which is subsequently fed into a meta-learner comprising machine learning classifiers for final sentiment prediction. This two-stage architecture is distinctive in its dual advantage: it not only leverages deep neural networks to understand linguistic intricacies but also capitalizes on traditional machine learning algorithms for enhanced classification robustness. The performance of the HPFCL model is rigorously evaluated using three widely adopted pre-trained word embeddings: GloVe, FastText, and Word2Vec. Across all embedding types, our model consistently outperforms conventional deep learning approaches, underscoring its effectiveness in sentiment classification tasks.

Furthermore, we investigate the role of large-language models (LLMs) in SA, particularly transformer-based models like BERT, RoBERTa, DistilBERT, ALBERT, ELECTRA, and XLNet. These models have revolutionized natural language processing (NLP) by effectively capturing long-range dependencies and contextual information through self-attention mechanisms. Our experiments on Amazon review datasets demonstrate that XLNet consistently outperforms other models in accuracy, with ELECTRA also showing competitive performance. XLNet's permutation language modeling approach, which predicts words based on their preceding context, allows it to capture more nuanced semantic relationships compared to BERT's masked language modeling. We also examine the effects of batch size and optimizer choice on model efficacy, providing insights into optimizing these parameters for improved results. For example, we found that using the Adam optimizer with a learning rate of 2e-5 and a batch size of 32 yielded the best performance for XLNet on our datasets.

This research contributes to the field of SA by proposing innovative methods and frameworks that enhance the accuracy and efficiency of sentiment classification in e-commerce. The findings offer practical implications for businesses seeking to leverage customer feedback for strategic decision-making and improved customer satisfaction. By combining traditional ML techniques with advanced DL and transformer-based models, we provide a comprehensive toolkit for analyzing customer sentiment at scale. Future work could explore the integration of multimodal data, such as combining text with images or user behavior data, to further enhance the robustness of sentiment analysis models in e-commerce applications.

1 Hypothesis and Research Issue

In the realm of e-commerce, it is hypothesized that employing advanced Sentiment Analysis (SA) techniques on customer data can significantly enhance data analysis processes, leading to improved understanding of customer preferences, behaviors, and satisfaction levels. This thesis proposes the integration of SA algorithms with traditional data analysis methods to extract deeper insights from customer feedback and interaction data. Leveraging Machine Learning (ML) and NLP techniques, this approach aims to provide

more accurate and actionable intelligence for e-commerce businesses to tailor their products, services, and marketing strategies effectively, thereby fostering enhanced customer experiences and increased business success.

1.1 Introduction

The prosperity of any business relies heavily on the quality of service delivered to its customers. Understanding buyer's perspectives regarding particular products is essential, as it enables businesses to evaluate tangible insights that are invaluable for growth in a fiercely competitive market. This, in turn, helps enhance product quality [1]. Social media has become a prevalent platform for individuals to share information and express their opinions about their experiences, largely due to advancements in internet technology [2]. Over time, the volume of reviews posted by users across various online platforms has increasingly reflected strong sentiments [3]. These reviews are valuable and can benefit both consumers and businesses. By examining these reviews, consumers can gain important insights into the quality of products. However, the overwhelming number of online reviews makes it challenging for individuals to manually read and analyze each evaluation to extract meaningful insights. Consequently, there is a growing demand for methods that enable computers to efficiently process and interpret this data, thereby reducing the need for human effort. To aid informed decision-making in e-commerce, various techniques are used to analyze customer reviews. Sentiment analysis (SA), in particular, identifies emotions in text comments to extract key insights that enhance purchasing decisions. SA emerges as a practical solution to tackle these challenges [4, 5]. SA is a vital and intricate task within the field of NLP. Its primary goal is to identify and extract subjective information from text, determining the sentiment polarity—whether it is neutral, negative, or positive—by utilizing text mining techniques and related methodologies.

The study of SA has explored various dimensions [6]. Among these, the detection of emotions and perspectives is commonly categorized into three levels: document-level, sentence-level, and aspect-level [7, 8, 9]. The first two levels pose interesting challenges, while the third involves more detailed and complex analysis. SA operates at three levels. Document-Level Sentiment Analysis (DSA) evaluates the overall sentiment of an entire document, categorizing it as positive or negative, with a focus on identifying and interpreting emotions through semantic understanding and sentence categorization [10]. Sentence-Level Sentiment Analysis (SSA) analyzes individual sentences to determine sentiment polarity, often used to extract user emotions and experiences from reviews [11, 12]. Aspect-Level Sentiment Analysis (ASA) delves deeper, identifying specific entities and features by extracting and classifying opinion words. For example, in the phrase "I love the Amazon program," the word "love" clearly conveys a positive sentiment, which is crucial for evaluating product attributes, determining polarity, and facilitating predictions [13, 14].

This thesis, we focus on the task of sentiment classification of e-commerce reviews, utilizing ML, DL, and transformer-based approaches. SA is crucial for understanding customer feedback, and e-commerce platforms provide a wealth of review data that can be analyzed for this purpose. To enhance the performance of our models, we have also employed and analyzed various embedding methods, such as Word2Vec, GloVe, and FastText. These embedding techniques play a vital role in representing textual data in a format suitable for computational models, thereby improving the accuracy of sentiment classification. The

dataset predominantly used for this research is the widely recognized Amazon Review Dataset [15], which is extensively utilized in SA studies due to its comprehensive and diverse review samples. This dataset provides a rich source of textual data that helps in training and testing various sentiment analysis models.

In this thesis, we present multiple SA models capable of accurately predicting the polarity of reviews (positive or negative). These models leverage ML, DL, and advanced transformer-based techniques, showcasing the evolution and effectiveness of different approaches in addressing sentiment classification challenges. The results and comparisons of these methodologies are systematically detailed in the subsequent chapters, providing insights into their performance and applicability in real-world e-commerce scenarios.

1.2 Sentiment analysis (SA) Approaches

SA methods can be categorized into three main types: Machine Learning (ML) approaches, Lexicon-Based approaches, and Transformer based approaches [16, 17, 18]. Machine learning approaches are the most widely adopted, utilizing algorithms and linguistic features to carry out sentiment classification. Lexicon-based approaches rely on sentiment lexicons, which consist of words and phrases commonly associated with positive or negative sentiments [19]. Hybrid approaches combine elements of both machine learning and lexicon-based methods to boost the effectiveness of sentiment analysis.

1.2.1 Machine learning approaches

Machine learning (ML) approaches include supervised, unsupervised, and semi-supervised learning. Supervised methods suit classification tasks with labeled data, while unsupervised methods identify patterns without predefined classes. Semi-supervised learning balances both, handling mostly unlabeled data with some labeled samples. Reinforcement learning optimizes rewards through trial-and-error [20]. Table 1 summarizes key challenges. Chapter 4 explores learning models with grid search hyperparameter tuning on the Amazon review dataset, using BoW, TF-IDF, Word2Vec, and average Word2Vec embeddings.

1.2.2 Deep learning approaches

Deep learning (DL) techniques, particularly Artificial Neural Networks (ANNs), are increasingly used for SA. Models like LSTMs, CNNs, and BiLSTMs learn meaningful representations without manual feature engineering [20].

Chapter 5 of this thesis explores sentiment classification using a 2D-CNN to capture spatial relationships in text. Additionally, a parallel-fused CNN-LSTM hybrid model is proposed, where CNN and LSTM extract features simultaneously. A Gradient Boosting Classifier then classifies sentiments, leveraging both models' strengths to improve performance.

1.2.3 Hybrid approaches

The hybrid approach merges lexicon-based methods with ML techniques, balancing lexical accuracy with ML's adaptability to context [21]. Sentiment lexicon scores serve as

input features for the classifier, leveraging both methodologies' strengths. This integration enhances sentiment analysis performance beyond standalone methods.

1.2.4 Transformer based approaches

Transformer-based models like BERT, XLNet, RoBERTa, and GPT have transformed sentiment classification by leveraging self-attention and pre-trained embeddings to capture context and semantics. Unlike traditional models, they learn hierarchical representations without manual feature extraction.

Chapter 5 explores these models for handling complex language patterns and subtle sentiment cues. Fine-tuning on sentiment datasets adapts their generalized knowledge to domain-specific tasks, enhancing classification performance.

1.3 Challenges in Sentiment Analysis Across Approaches

The challenges of all these approaches are listed in Table 1.1. Each method has its own difficulties, such as the high computational needs of transformer models, the requirement for large amounts of labeled data in supervised methods, and the complexity of combining features in hybrid models. Common issues like data imbalance, lack of interpretability, and scalability also affect the performance and application of sentiment classification models.

Approach	Challenges
Machine Learning	- Difficult feature engineering - Data imbalance - Sensitive to noisy data - Scaling to large data - Expensive hyperparameter tuning
Deep Learning	- Requires large datasets - Hard to interpret - Prone to overfitting - Long training times - Resource-intensive
Hybrid Approaches	- Complex integration - Difficult optimization - Data consistency issues - Balance between accuracy and interpretability
Transformers	- High computational requirements - Large model sizes - Data dependency for fine-tuning - Limited interpretability of predictions - Challenges in domain-specific adaptation

Table 1: Challenges in Sentiment Analysis Approaches

1.4 Literature Review

We categorize related work into lexicon-based, ML-based, DL-based, and transformer-based approaches, each contributing uniquely to sentiment analysis. Lexicon-based methods use predefined dictionaries, while ML and DL techniques enhance adaptability. This section reviews recent literature, illustrating the evolution of these methods in analyzing Amazon reviews.

1.4.1 Sentiment analysis using Lexicon based techniques

In this section, recent papers from the last four years on Amazon reviews utilizing lexicon-based approaches are presented in Table 2.

S. No.	Author	Publication	Year	Lexicon-based Technique	Dataset	Optimal performance
1	Shobayo et al.[22]	MDPI (Analytics)	2024	VADER	Amazon Fashion review	for positive sentiment: 0.91, negative sentiment: 0.93
2	Ullah et al.[23]	MDPI (Computers)	2023	QLeBERT	Amazon product reviews	F1-macro = 0.91
3	ElKafrawy et al.[24]	IEEE	2022	TextBlob	Amazing Egypt's electronics (scraped data)	Accuracy (BERT): 91%, (TextBlob): 65%
4	Welgamage et al.[25]	IEEE	2022	VADER	Amazon electronic reviews	Accuracy (BERT): Future work mentions improvements
5	Catelli et al.[26]	MDPI (Electronics)	2022	NooJ-based lexical analysis	Multiple Italian websites (ciao.it, amazon.it, etc.)	Accuracy, F1-Score
6	Mukherjee et al.[27]	Elsevier (Procedia Computer Science)	2021	SentiWordNet	Amazon cell phone reviews	Accuracy, Precision, Recall, F1-Score
7	Li et al.[28]	IEEE	2021	SentiWordNet, WordNet	Amazon Musical Instruments Reviews	Accuracy, Precision, Recall, F1-Score
8	Kim et al.[29]	IEEE (Engineering Management Review)	2021	Harvard University's GI, LM Dictionary	Smartphone reviews on Amazon.com	Accuracy, Precision, Recall, F1-Score
9	Ouatiti et al.[30]	Springer (Computational Vision and Bio-Inspired Computing)	2020	TextBlob	Amazon book reviews	Lexicon-based

Table 2: Comparison of Lexicon-based Techniques

1.4.2 Sentiment analysis using Machine learning techniques

ML techniques like SVM, Naïve Bayes, and Random Forests have been widely used in SAAR, with ensemble learning improving performance. While effective and interpretable, they struggle with large-scale data and nuanced sentiment patterns, making DL and transformer models more competitive. However, they remain valuable baselines in SAAR studies. Table 3 lists relevant works.

S. No.	Author	Publication	Year	Machine learning Technique	Dataset	Optimal performance
1	Myasar R. Tabany, Meriem Gueffal [31]	Journal of Advances in Information Technology	2024	SVM, Random Forest, Naive Bayes, Logistic Regression	Books, Gift card, Digital software, Electronics	Accuracy-88%
2	A. Semary et al. [32]	PLOS ONE	2024	Random Forest	Amazon musical instrument reviews	Accuracy -99%
3	Haili Wang [33]	Highlights in Science, Engineering and Technology	2024	SVM	Amazon fine food reviews	Accuracy- 89.67%
4	Arman Sarraf et al. [34]	International Journal of Computer (IJC)	2024	Random Forest, Logistic Regression, Decision Tree	Amazon food reviews	Accuracy (Logistic Regression)- 90%
5	Aakash et al. [35]	Procedia	2024	SVM, Random Forest, Naive Bayes, Logistic Regression	Amazon product review	Accuracy (GRU)-87%
6	Vipin Jain et al. [36]	IEEE	2024	SVM, Random Forest, Decision Tree, XGBoost, Ensembled method	Amazon product reviews	F1 score -91%
7	Ameer Hamza et al. [37]	Bulletin of Business and Economics	2024	SVM, Random Forest, Naive Bayes, Logistic Regression	Amazon Electronics category dataset	Accuracy (BERT + TF-IDF) -82.65%
8	Sangeetha J, Kumaran U [5]	Measurement: Sensors	2023	SVM, Random Forest, Naive Bayes, Decision Tree	Amazon user review dataset	Accuracy- 95.8%
9	TR Aravindan et al. [38]	Recent Developments in Electronics and Communication Systems	2023	Naive Bayes, Logistic Regression, Linear SVC	Amazon fine food reviews	Accuracy (Logistic Regression)-87.61%

Table 3: Summary of Machine learning techniques used on amazon review sentiment analysis.

1.4.3 Sentiment analysis using Deep Learning techniques

SAAR has seen significant advancements through the use of DL techniques. DL methods, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), Gated Recurrent Units (GRUs), and Bi-directional LSTMs (BiLSTMs), have enabled researchers to extract high-level features from text data, resulting in improved sentiment classification performance. These approaches leverage the power of neural networks to capture complex patterns and dependencies within text, making them particularly effective for large and diverse datasets such as those from Amazon. Table 4 and Table 5 provides a list of papers that have utilized this approach.

S. No.	Author	Publication	Year	DL-based Technique	Dataset	Optimal Performance
1	Haili Wang [33]	Highlights in Science, Engineering and Technology	2024	CNN, BiLSTM	Amazon fine food reviews	Accuracy - 89.67%
2	Aakash et al. [35]	Procedia Computer Science	2024	LSTM, GRU	Amazon product review	Accuracy - 87% (GRU)
3	Kusal et al. [39]	International Conference on Artificial Intelligence on Textile and Apparel	2023	LSTM, BiLSTM	Amazon reviews dataset	RoBERTa - Accuracy - 80.28%
4	Sangeetha et al. [5]	Measurement: Sensors	2023	RNN-LSTM	Amazon user review dataset	95.8% Accuracy, 95.4% Precision, 95.6% Recall, 95.2% F-measure
5	Qorich et al. [40]	The Journal of Supercomputing	2023	CNN	Amazon reviews dataset	Accuracy - 90%
6	Mahadevaswamy [41]	Procedia Computer Science	2023	BiLSTM	Amazon product review	Accuracy - 91.4%
7	Atandoh et al. [42]	Journal of King Saud University – Computer and Information Sciences	2023	CNN, LSTM	Amazon reviews	95% Accuracy
8	Agrawal et al. [43]	Measurement: Sensors	2023	Multi Layer Perceptron	Amazon product review	96% Accuracy
9	Kim et al. [44]	Electronic Commerce Research and Applications	2023	Deep Auto Encoder	Amazon recliner product review	N/A
10	Hájek et al. [42]	Journal of Business Research	2023	CNN, DNN, DNN n-gram	Amazon fake reviews	Accuracy-69.22%
11	Mutinda et al. [45]	Applied Sciences	2023	CNN	Amazon product reviews	F-measure - 88.73%
12	Zhang et al. [46]	Sensors	2023	BGRU Attention	Amazon product review	Accuracy-62.57%
13	Iftikhar et al. [42]	TELKOMNIKA	2023	CNN, LSTM	Various Amazon products	Accuracy-95.17%
14	N. Habbat et al. [47]	IEEE	2023	CNN, LSTM, BiLSTM, GRU	French Amazon Customer Reviews	N/A

Table 4: Summary of Deep Learning based Techniques in SAAR

24	Bhuvaneshwari et al. [48]	Springer	2022	BiLSTM	Amazon Cell Phone and Accessories	Not mentioned
25	Chamekh et al. [49]	Springer	2022	LSTM	Smartphone Product Reviews	Accuracy - 95%
26	Gondhi et al. [50]	Computational Intelligence and Neuroscience	2022	LSTM	Amazon Review Dataset 2018	Not mentioned

Table 5: Continuation of Summary of Deep Learning Based Techniques in amazon review sentiment analysis

1.4.4 Sentiment analysis usinf Transformer Based techniques

TB models, such as BERT, RoBERTa, and XLNet [51], have revolutionized the field of NLP, offering state-of-the-art performance in SAAR. These models utilize self-attention mechanisms to capture long-range dependencies in text, leading to superior contextual understanding compared to traditional DL approaches. TB models have proven highly effective in sentiment classification, especially for large-scale datasets like Amazon product reviews, where sentiment nuances can be subtle and context-dependent.

S. No.	**Author**	**Publication**	**Year**	**TB-based Technique**	**Dataset**	**Optimal Performance**
1	Shobayo, Olamilekan et al. [22]	Analytics (MDPI)	2024	BERT	Amazon Fashion review dataset	GooglePaLM: Correct +ve 0.91 and Correct -ve 0.93
2	Hamza, Ameer et al. [37]	Bulletin of Business and Economics	2024	BERT + TF-IDF Vectorizer	Amazon Electronics category dataset	Accuracy: 0.8265
3	Kusal, Sheetal et al. [39]	Int. Conf. on AI on Textile and Apparel (Springer)	2024	BERT, RoBERTa, Distil-RoBERTa	Amazon reviews dataset	RoBERTa Accuracy: 80.28%
4	Atandoh, Peter et al. [52]	Journal of King Saud Univ. – Computer and Information Sciences (Elsevier)	2023	BERT	Amazon reviews	Accuracy-95%
5	Zhang, Xiangsen et al. [46]	Sensors (MDPI)	2023	BERT + Bi-GRU	Amazon product review	Accuracy- 74.37%
6	Iftikhar, Saman et al. [53]	TELKOMNIKA	2023	BERT	Amazon products	BERT performs best
7	Ullah, Aman et al. [23]	Computers (MDPI)	2023	BERT	Amazon product reviews	F1-Macro: 91% (QLe-BERT)
8	Rangarajan, Prasanna et al. [54]	IEEE	2023	BERT	Amazon product reviews	Accuracy-92.25%
9	Ali, Mohammed AM et al. [55]	Journal of Harbin Engineering Univ.	2023	BERT, BiLSTM, DCNN	Amazon product reviews	Accuracy- 94%
10	Monsoor, Razin Sumyta et al. [56]	IEEE	2023	BERT, RoBERTa	Amazon reviews	RoBERTa: Accuracy 82%, F1-Score-90.1%

Table 6: Summary of Transformer Based Techniques for amazon review sentiment analysis

2 Objectives

In this thesis, we focus on developing effective SA models. To achieve this goal, we set out to address a set of specific research objectives that guide the exploration and implementation of various techniques in SA. The following objectives outline the key areas of this work:

- Develop a comprehensive sentiment analysis framework for products on e-commerce platforms.
- Classification of user opinion into different sentiment categories using Machine Learning algorithms.
- Investigation of sentiment analysis on e-commerce datasets using deep learning approaches.
- To fine-tune the XLNet model for Amazon review sentiment analysis and conduct a comparative evaluation of its performance against other transformer-based models.

3 Proposed Methods

This thesis explores advanced SA techniques for e-commerce, integrating ML, deep learning, and transformer models. By leveraging pre-trained embeddings, attention mechanisms, and hyperparameter optimization, we enhance sentiment classification accuracy. The proposed methods address challenges in analyzing large-scale reviews and are detailed in the following subsections.

3.1 Hyperparameter Optimization of Machine Learning Models using grid search for Amazon Review Sentiment Analysis

In this work, we used grid search to process hyperparameter optimization to improve the parameters of six machine learning models: Decision Tree(DT), Logistic Regression(LR), Naive Bayes(NB), Support Vector Machine(SVM), XGBoost(XG), and Random Forest(RF). The principal aim of this research is to ascertain the most suitable or ideal values for the hyperparameters that result in the most superior model while taking into consideration the unpredictability of the computation costs during training. Additionally, it involves assessing machine learning models using varying hyperparameter values. Building accurate and effective machine learning models for Amazon sentiment analysis requires tuning hyperparameters. In this approach, a selected set of algorithms is employed, wherein a group of hyperparameters is specifically defined, and optimal hyperparameter values, which optimize the model's performance on a validation set, are determined by means such as grid search. The refinement of hyperparameters can augment the precision and generalization capabilities of machine learning models for Amazon sentiment analysis.

In this research work, we aim to perform sentiment analysis on user reviews from the Amazon Women's clothing E-commerce dataset. We employed four feature extraction methods, including Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Average Word2Vec, and the combination of TF-IDF with Word2Vec. To convert the textual input into numerical form, various feature extraction techniques were

employed. To evaluate their performance, we utilized the proposed Grid Search Hyperparameter Optimisation (GSHO) method to obtain the optimal hyperparameters for each ML model. This involved conducting a grid search among numerous hyperparameters and selecting the ones that produced the best results. We used accuracy, precison, recall anf F1-score to evaluate our models. In addition to this, we used the AUC metric and presented a confusion matrix for each classifier. For Naive Bayes, we used BoW and TF-IDF feature extraction techniques only. The findings of this research work suggest that the optimization of model hyperparameters has a noteworthy and statistically significant effect on the performance of the machine learning models.

3.1.1 Proposed methodology

The suggested methodology for Amazon reviews sentiment analysis is introduced in this section. Our proposed methodology's flow diagram is illustrated in figure 1. The evaluated Amazon women's clothing review data undergoes a number of preprocessing steps, including feature extraction, during which relevant information is extracted from the raw data. The vectors obtained from feature extraction are then used as inputs to the machine learning (ML) models for training and testing. To enhance the effectiveness of the model, the grid search approach is utilized for parameter tuning. This section presents the proposed model, including preprocessing, feature extraction, and hyperparameter optimization techniques in their entirety.

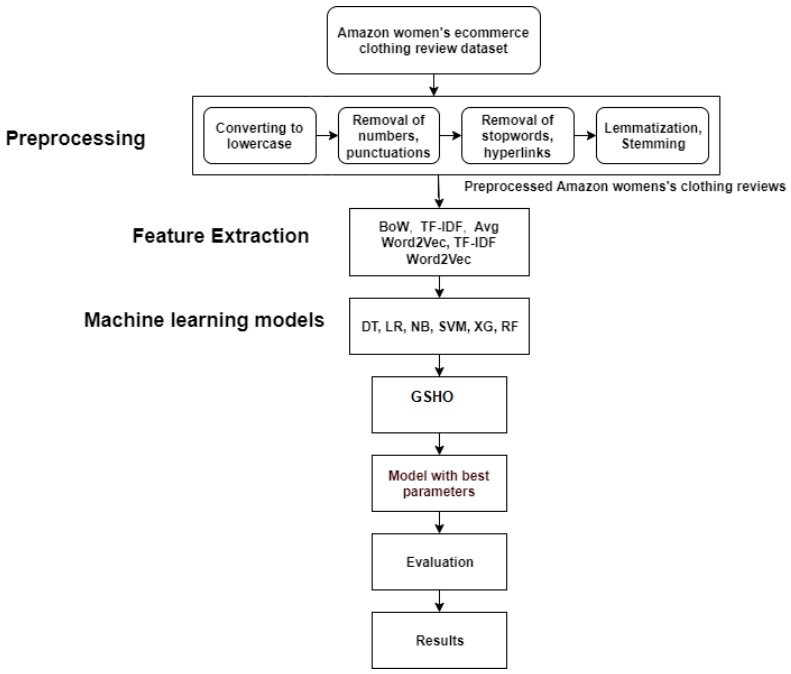


Figure 1: Workflow of the suggested approach for fine-tuning hyperparameters of ML models for SA of Amazon Women's Clothing E-Commerce reviews.

Table 7: Hyperparameter values for different machine learning models

Model	Hyperparameter forms
DT	'max_depth': [1, 5, 10, 50, 100, 500, 1000], 'min_samples_split: [5, 10, 100, 500]
LR	'penalty': ['l1','l2'] 'C':[0.0001,0.001,0.01,0.1,1,10,100,1000]
NB	'alpha':[0.0001,0.001,0.01,0.1,1,10,100,1000,10000]
SVM	'alpha' : [0.0001,0.001, 0.01, 0.1, 1, 10, 100, 1000] 'penalty' : ['l1', 'l2']
XG	'no_of_estimators' :[5, 10, 50, 100, 120] 'max_depth':[1, 5, 7, 10, 15, 25, 30]
RF	'no_of_estimators' : [5, 10, 50, 100, 120] 'max_depth' :[1, 5, 7, 10, 15, 25, 30]

3.1.2 Experiments

3.1.2.1 Experimental setup

The dataset is divided into three separate subsets, namely the training set, cross-validation set, and testing set. The input data is the review text column and the output data is the recommended IND column. During the initial data partitioning step, a training set and a testing set are created, with typically around 30% of the original dataset being reserved for testing purposes. In the second step of data partitioning, the initial training set is divided into two subsets: a new training set and a cross-validation set. Around 30% of the original training data is allocated for the purpose of cross-validation. The four feature extraction approaches are then used to turn the reviews into vectors, which are then input into the machine learning models. Table 7 displays each ML algorithm's hyperparameter forms.

3.1.3 Results analysis

The table 8 shows the results of DT model using different feature representations (BoW, TF-IDF, AvgW2V, TF-IDF W2V) with different hyperparameters (min_samples_split, max_depth) and the hyperparameter values used followed by performance metrics. The optimum set of hyperparameters that optimise the cross-validation AUC score is then found using an exhaustive search with GridSearchCV. The number of cross-validation folds is 3. The model's capability to distinguish between positive and negative classes is evaluated using the ROC curve and AUC values as well as confusion matrix is also used. These counts provide valuable insights into the model's performance in terms of its accuracy in identifying and categorizing data points, and aid in assessing the overall classification efficacy of the model.

In table 8 the BoW model has the lowest F1-score of 0.893 and AUC of 0.817 among all the models. The AvgW2V model has the highest recall of 0.947 and F1-score of 0.901 among all the models. It also has the highest AUC of 0.836. Overall, we observe that the AvgW2V model performs the best among all the models. Confusion matrices for DT with four feature extraction techniques is shown in fig 2. Area under the curve (AUC) for DT

Vector	Algorithm	Min. Sample Splits	Max Depth	Accuracy	Precision	Recall	F1-score	AUC
BoW	DT	500	100	0.823	0.884	0.903	0.893	0.817
TF-IDF	DT	500	50	0.832	0.882	0.916	0.899	0.796
AvgW2V	DT	500	50	0.829	0.858	**0.947**	**0.901**	**0.836**
TF-IDF W2V	DT	500	10	0.829	0.864	0.939	0.900	0.821

Table 8: DT model performance on the Women's Clothes Reviews dataset using GSHO with different feature extraction techniques.

with four feature extraction techniques is shown in fig 3 In table 9, the best-performing

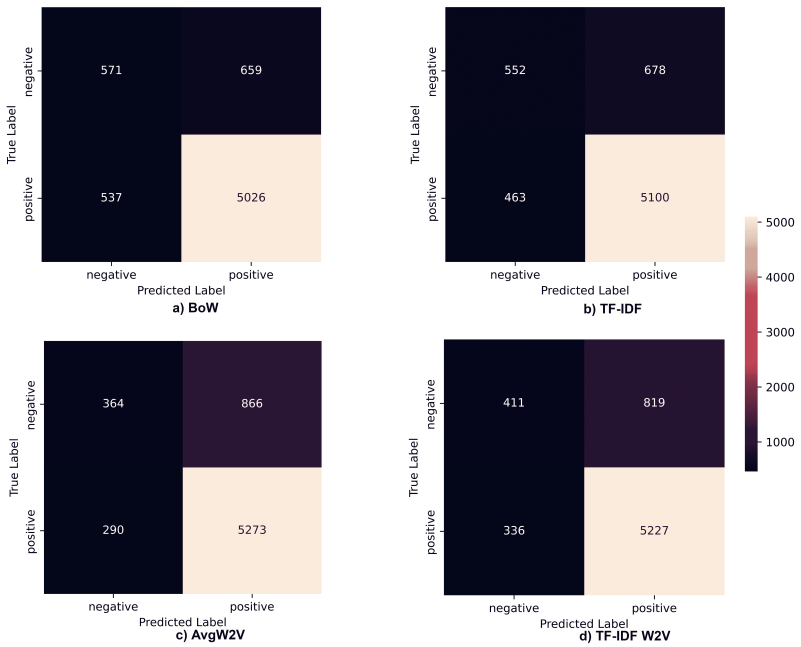


Figure 2: Confusion matrices for DT

Vector	Algorithm	Regularization	C	Accuracy	Precision	Recall	F1-score	AUC
BoW	LR	L2	0.1	**0.892**	0.910	**0.963**	**0.936**	**0.931**
TF-IDF	LR	L2	10	0.890	**0.916**	0.952	0.934	0.930
AvgW2V	LR	L2	10	0.875	0.895	0.960	0.926	0.899
TF-IDF W2V	LR	L1	1	0.868	0.888	0.960	0.922	0.882

Table 9: LR model performance on the Women's Clothes Reviews dataset using GSHO with different feature extraction techniques.

technique was Bag-of-Words (BoW) with L2 regularization and hyperparameter C=0.1, achieved a good accuracy value of 0.892 and an F1-score of 0.936. The worst-performing technique was TF-IDF weighted Word2Vec with L1 regularization and hyperparameter C=1, with accuracy of 0.868 and an F1-score of 0.922. Both BoW and TF-IDF performed well. However, BoW achieved a slightly better F1-score and AUC than TF-IDF. For the grid search, we employ a logistic regression model with a "liblinear" solver as the estimator. The AUC and the confusion matrix of LR for various feature extraction techniques is given in figures 5 and 4 respectively. We can apply naive bayes only on positive values so we have used only BoW and TF-IDF. The table 10 presented contrasts two extraction techniques, BOW and TF-IDF, both utilizing Naive Bayes as the algorithm with a hyperparameter-alpha value of 0.1. The outcomes reveal that the NB model performed better using BoW

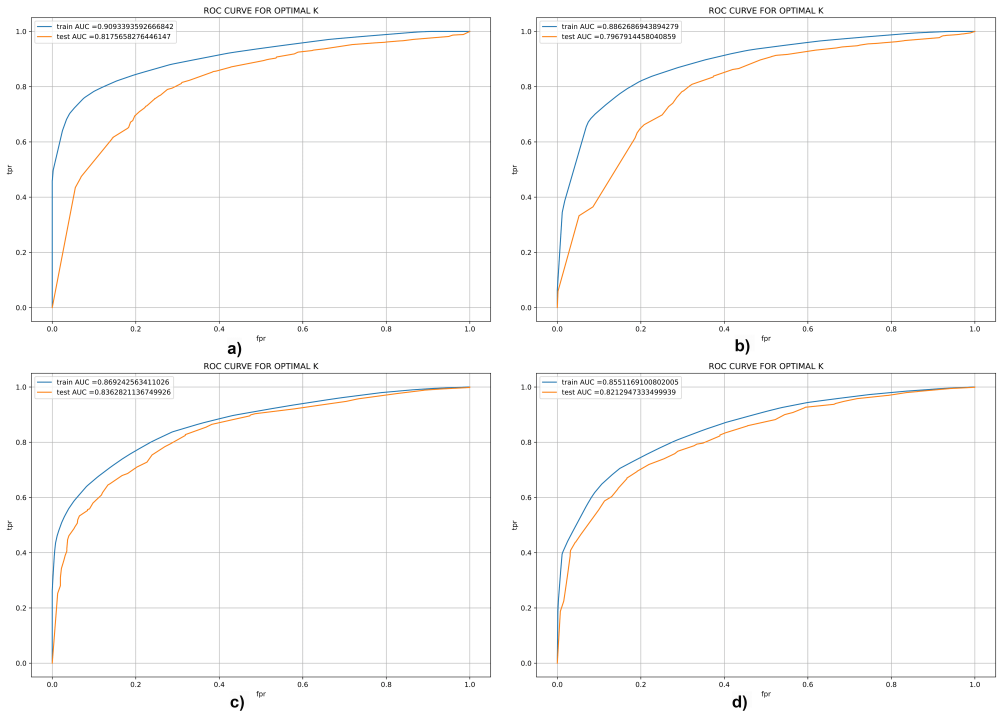


Figure 3: AUC for DT a) BoW b) TF-IDF c) Average Word2Vec d) TF-IDF Word2Vec

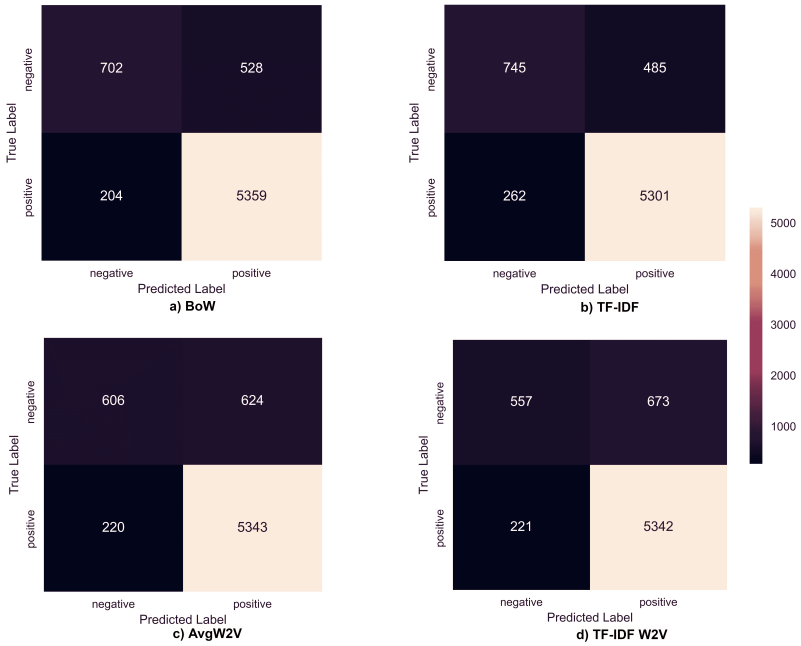


Figure 4: Confusion matrices for LR

with highest accuracy of 0.884, precision of 0.939, recall of 0.918 and F1-score of 0.928 compared to TF-IDF as the feature extraction technique. It makes use of a multinomial naive bayes model. Using 5-fold cross-validation, it first carries out hyperparameter tuning to find the optimum alpha (the smoothing parameter) value.

The AUC and the confusion matrix of NB for various feature extraction techniques is given in figures 7 and 6 respectively. In table 11 The SVM algorithm was trained with L1 or

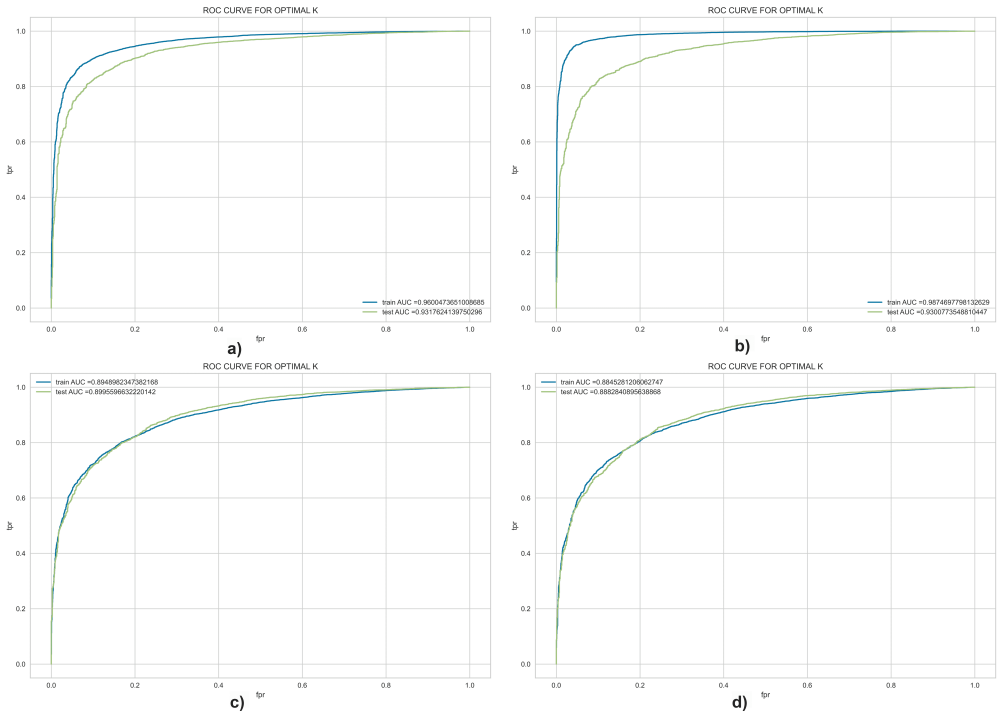


Figure 5: AUC for LR a) BoW b) TF-IDF c) Average Word2Vec d) TF-IDF Word2Vec

Vector	Algorithm	Alpha	Accuracy	Precision	Recall	F1-score	AUC
BoW	NB	0.1	**0.884**	**0.939**	0.918	**0.928**	0.918
TF-IDF	NB	0.1	0.859	0.860	**0.987**	0.919	**0.921**
AvgW2V	NB	—	—	—	—	—	—
TF-IDF W2V	NB	—	—	—	—	—	—

Table 10: NB model performance on the Women's Clothes Reviews dataset using GSHO with different feature extraction techniques.

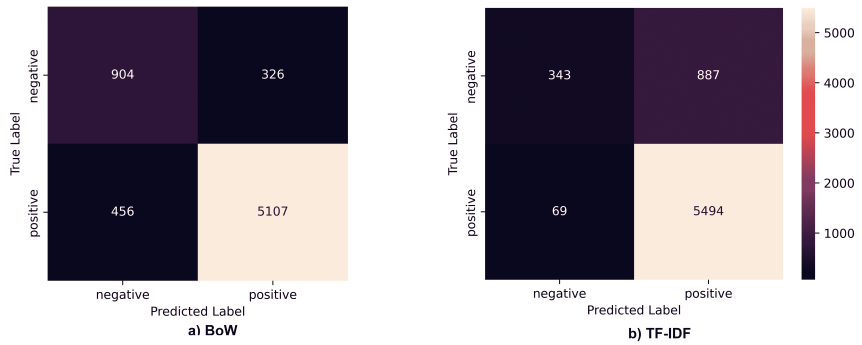


Figure 6: Confusion matrices for NB

L2 regularization and different values of alpha. Based on the results, the best performing model is the one using TF-IDF with L2 regularization and an alpha value of 0.0001. This model achieved an accuracy of 0.897, precision of 0.916, recall of 0.962, F1-score of 0.938, and a AUC of 0.937. The BOW model also performed well with an accuracy of 0.889, precision of 0.907, recall of 0.963, F1-score of 0.934, and a AUC of 0.927. The AvgW2V

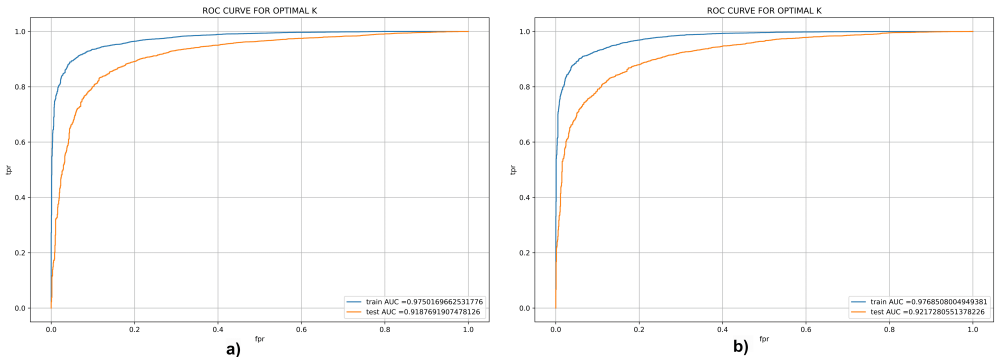


Figure 7: AUC for NB a) BoW b) TF-IDF

and TF-IDF W2V models did not perform as well as the other two models. The AUC and CM of SVM are shown in Fig.9 and Fig.8.

Vector	Algorithm	Regularization	Alpha	Accuracy	Precision	Recall	F1-score	AUC
BoW	SVM	L2	0.001	0.889	0.907	0.963	0.934	0.927
TF-IDF	SVM	L2	0.0001	**0.897**	**0.916**	**0.962**	**0.938**	**0.937**
AvgW2V	SVM	L1	0.0001	0.874	0.892	0.962	0.926	0.898
TF-IDF W2V	SVM	L1	0.0001	0.867	0.884	0.964	0.922	0.889

Table 11: SVM model performance on the Women's Clothes Reviews dataset using GSHO with different feature extraction techniques.

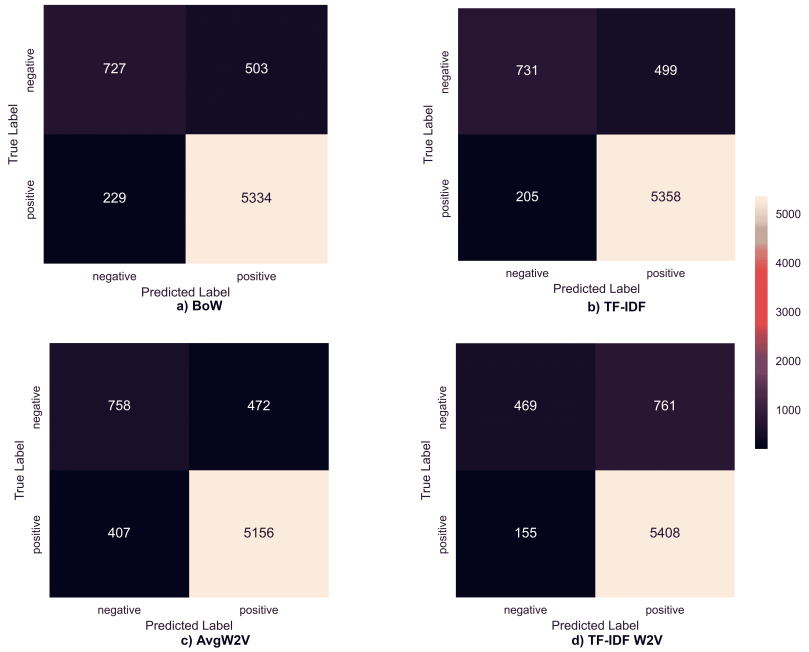


Figure 8: Confusion matrices for SVM

The hyperparameters in XG boost in table 12 that are taken into account are the maximum depth and the number of elements. The hyperparameter's number of elements was maintained at 120. The maxdepth value for the trees in the hyperparameter varied for each feature representation. In this experiment cross validation is set to 3. we observe

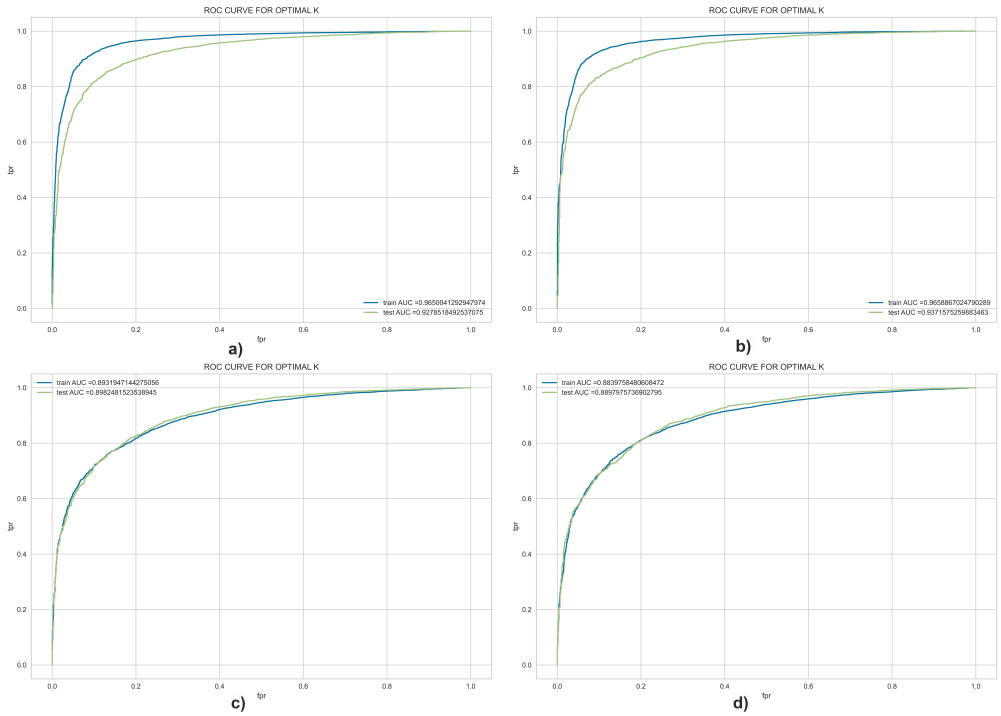


Figure 9: AUC for SVM a) BoW b) TF-IDF c) Average Word2Vec d) TF-IDF Word2Vec

Vector	Algorithm	No of Elements	Max Depth	Accuracy	Precision	Recall	F1-score	AUC
BoW	XG	120	10	**0.888**	**0.908**	**0.960**	**0.933**	**0.923**
TF-IDF	XG	120	7	0.881	0.901	0.960	0.929	0.921
AvgW2V	XG	120	15	0.859	0.887	0.948	0.917	0.878
TF-IDF W2V	XG	120	25	0.856	0.882	0.950	0.915	0.876

Table 12: XG model performance on the Women's Clothes Reviews dataset using GSHO with different feature extraction techniques.

that BOW-based model achieved the highest accuracy of 0.888 and F1-Score of 0.933, indicating that it performs well in terms of correctly classifying positive and negative reviews. It also has high Precision of 0.908 and recall of 0.96. The TF-IDF based model also performed well with an accuracy of 0.881 and an F1-Score of 0.929. However, its precision is 0.901 and recall is 0.96 values are slightly lower than that of the BOW-based model. In contrast, the Average Word2Vec and TF-IDF Weighted Word2Vec models have lower performance with an F1-Score of 0.917 and 0.915, respectively. The AUC and CM of XG Boost are presented in fig.11 and fig.10. The hyperparameters in Random Forest in table

Vector	Algorithm	No of Elements	Max Depth	Accuracy	Precision	Recall	F1-score	AUC
BoW	RF	100	25	0.879	0.933	0.919	0.926	0.918
TF-IDF	RF	120	30	**0.880**	0.922	0.932	**0.927**	**0.919**
AvgW2V	RF	120	25	0.852	0.867	0.967	0.914	0.869
TF-IDF W2V	RF	120	25	0.849	0.864	0.968	0.913	0.864

Table 13: RF model performance on the Women's Clothes Reviews dataset using GSHO with different feature extraction techniques.

13 are identical to those in the XG boost model. CV is set to 5. Among the considered methods for feature extraction, the TF-IDF technique yielded the most favorable results in terms of accuracy for the test data, exhibiting a value of 0.880 with F1-score of 0.927.

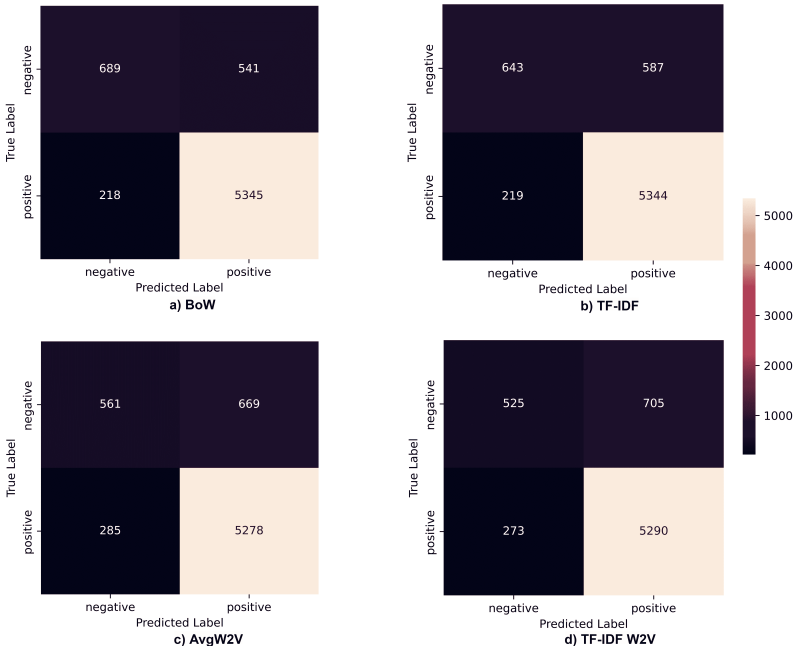


Figure 10: Confusion matrices for XG

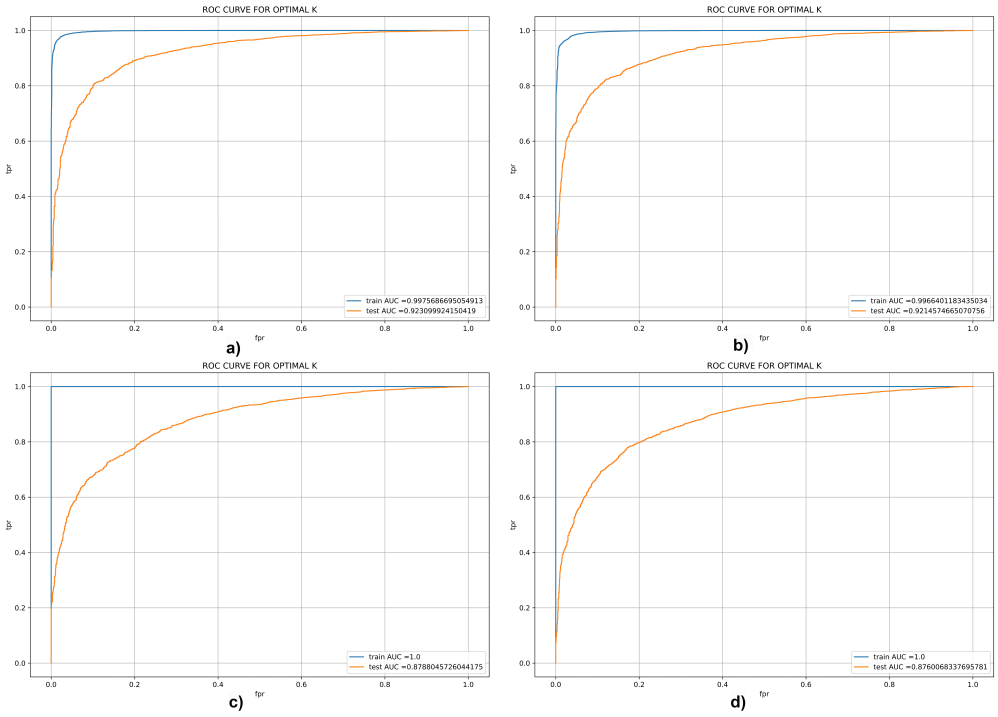


Figure 11: AUC for XG a) BoW b) TF-IDF c) Average Word2Vec d) TF-IDF Word2Vec

Conversely, the TF-IDF Word2Vec approach delivered the least accurate results, with a recorded value of 0.849. The AUC and CM of RF are shown in fig13 and fig12 respectively.

After conducting a study on several machine learning models such as DT, LR, NB, RF, SVM, and XG, it was observed that the BoW feature extraction technique provided the best results in LR, NB, and XG models. However, the TF-IDF technique was found to be more effective in the RF and SVM models. The Average word2vec got a good result

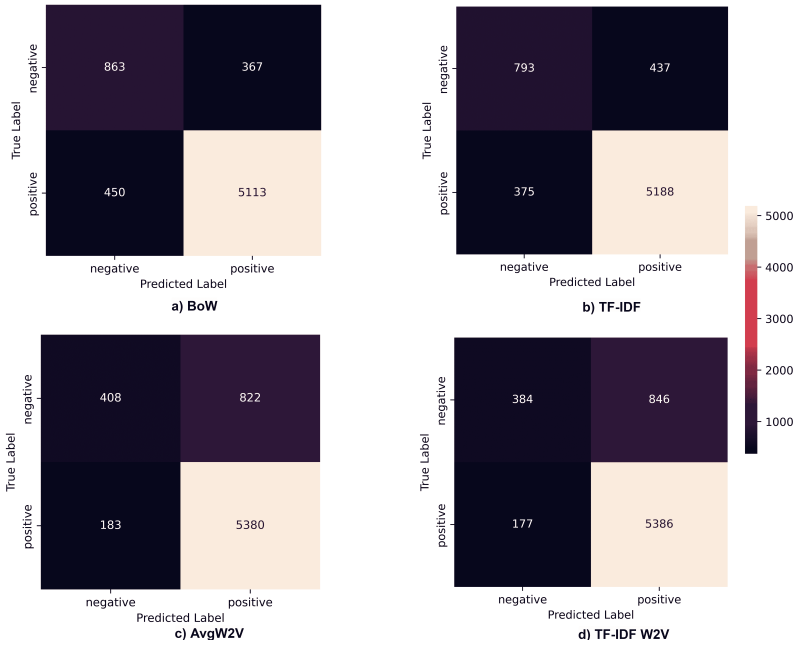


Figure 12: Confusion matrices for RF

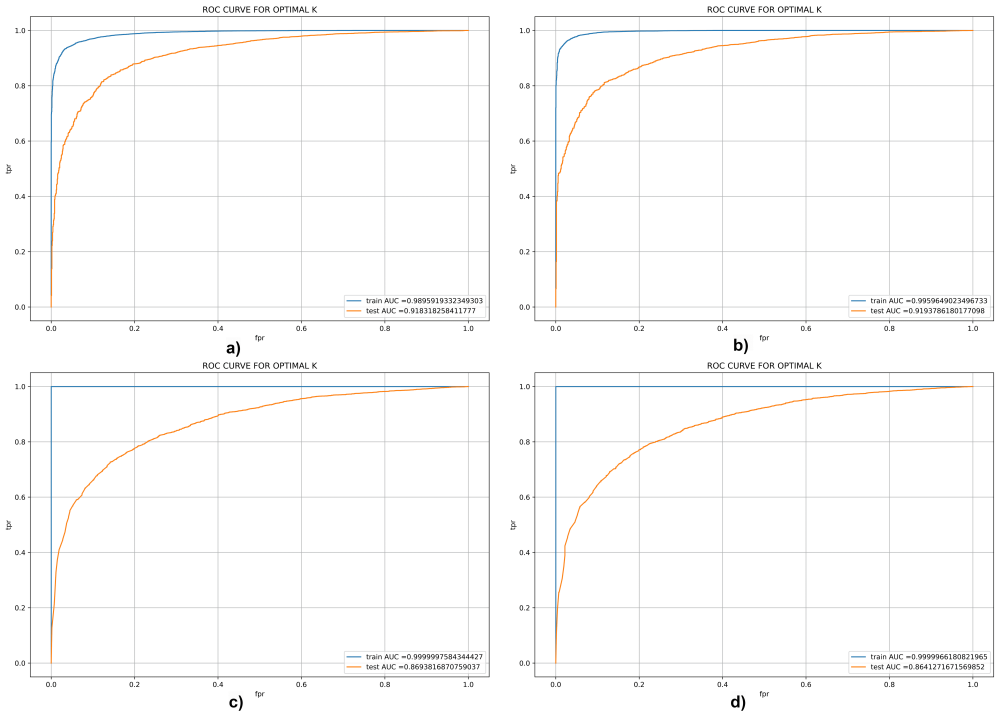


Figure 13: AUC for RF a) BoW b) TF-IDF c) Average Word2Vec d) TF-IDF Word2Vec

in DT model and TF-IDF word2vec technique did not yield satisfactory results in any of the models. In conclusion, the BoW and TF-IDF techniques were found to be the most effective feature extraction techniques for our study. It is also worth noting that the imbalanced nature of the dataset resulted in lower accuracy compared to other metrics. Therefore, accuracy may not be a reliable metric for evaluating the performance of a model when dealing with an imbalanced dataset. Precision and recall are better metrics

to use as they focus on the performance of the model for the minority class, which is often more important in real-world applications. In summary, based on the study conducted, we recommend using BoW and TF-IDF feature extraction techniques and using precision and recall as evaluation metrics when dealing with an imbalanced dataset.

3.2 Unleashing the Power of 2D CNN with Attention and Pre-trained Embeddings for Enhanced Online Review Analysis

Convolutional Neural Networks (CNNs), widely used in image processing, can also be applied to text analysis. While 1D CNNs capture local word-level features, 2D CNNs with attention (Att+2D CNN) can learn multi-dimensional patterns in text. Attention layers help retain crucial information, especially when processing long sequences.

We propose an Att+2D CNN model for extracting insights from Amazon cell phone and Kindle reviews. We use pretrained word embeddings (GloVe and Word2Vec) as inputs and compare our model with other DL architectures, including 1D CNN, 2D CNN, LSTM, BiLSTM, GRU, and BiLSTM with attention, across both balanced and unbalanced datasets.

3.2.1 Methodology

The process begins with obtaining the datasets from Kaggle, followed by data annotation. The resulting annotated dataset is then preprocessed to eliminate noise and redundant data. Next, feature extraction is performed using pre-trained word embeddings. The extracted features are then utilized as inputs to the existing DL models and the proposed model , which are subsequently evaluated based on various performance metrics. The architecture of the proposed model is shown in Fig 14.

3.2.2 Proposed model

We propose a novel deep learning model designed for the purpose of text classification tasks, which leverages a synergistic combination of convolutional neural networks (2D CNN) and attention mechanisms. The architectural design of our model entails an initial embedding layer, followed by multiple convolutional layers, each employing distinct filter sizes 1,2,3,5 and 7. The hyperparameters used in the model are shown in table 14. The input data is represented as a 2D matrix, where each row corresponds to a sequence of words or tokens, and each column represents the embedding dimensions of the words. These convolutional layers efficiently capture localized features within the text by performing convolutions using filters of varying widths across the embedded input. Subsequently, we employ max pooling to extract the most salient features from each convolutional layer. These extracted features are then concatenated and passed through an attention layer, wherein attention weights are computed by means of a softmax function applied to a learned context vector. The attention weights play a pivotal role in modulating the pooled features, thereby emphasizing the most significant information for accurate classification. The attended features undergo further processing via a dense layer, followed by flattening and dropout regularization. Finally, we employ a sigmoid activation function to produce the predicted probability of the input belonging to a specific class. The model is trained using the binary cross-entropy loss function, with optimization carried out through the Adam optimizer. Empirical results serve to illustrate the remarkable

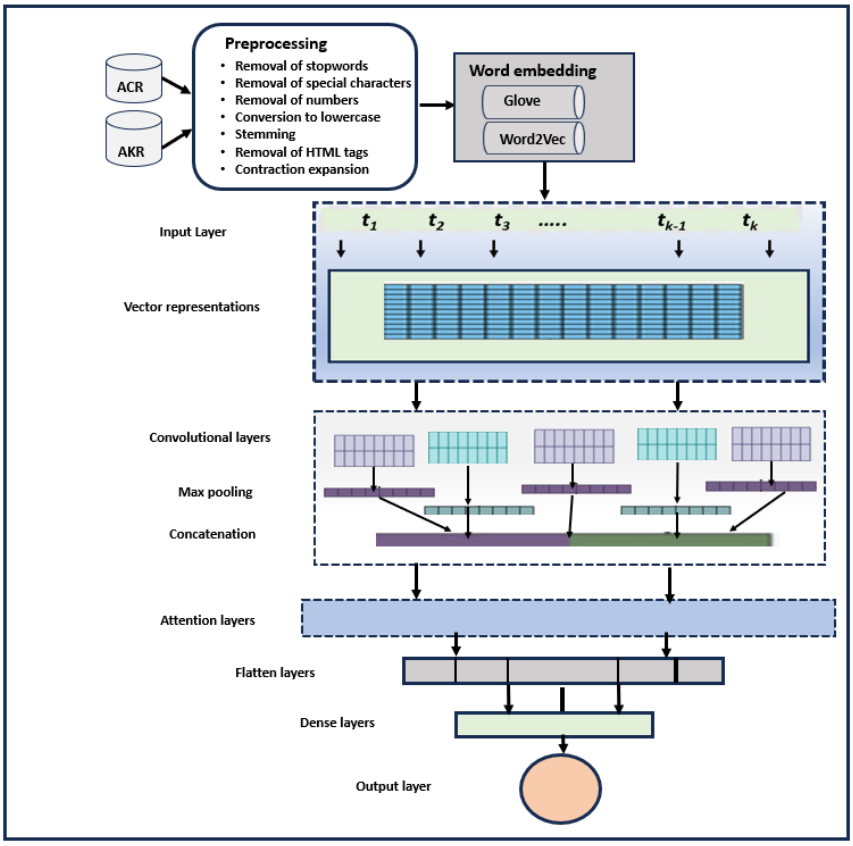


Figure 14: Architecture of the proposed model.

efficacy of our proposed model in achieving superior levels of accuracy when employed in diverse text classification tasks.

3.2.2.1 Attention layer

Attention has demonstrated its efficacy in numerous tasks within Natural Language Processing (NLP). There are two types of attention mechanisms: Additive attention and scaled dot-product attention. In our work, we specifically utilized the scaled dot-product attention mechanism [57]. Instead of adding the input representations together, we calculated the similarity or dot product between pairs of representations. This similarity serves as the attention weight, determining how much attention each input representation should receive. The architecture of the attention mechanism is showed in Fig 15.

Polarity score and attention score are fundamental concepts when analyzing neural classification models with attention mechanisms. The polarity score denotes the estimated sentiment or polarity of an input sequence, discerning whether it is positive or negative. Conversely, the attention score evaluates the importance and pertinence of individual word tokens within the sequence [57].

h_1, h_2, ..., h_j, ..., h_n represents a sequence of token representations in the given text. Each token is denoted as h_j, where the subscript j indicates the position of the token within the sequence. These tokens are fed to attention layer. Attention scores are calculated by the equation 1.

$$h = \sum_j \alpha_j h_j \tag{1}$$

where $\sum$ denotes the summation operation by adding up all the attention scores α_j and token representation h_j in the sequence.

$$s = h^\top W \tag{2}$$

Finally the polarity score s which is linear layer output, represented by equation 2 is obtained by taking the dot product of input value h and the transposed weight vector $^\top W$. The sigmoid function, denoted as $\sigma(x)$, is mathematically represented in equation 3 as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

where x is the input value. it maps the input to the range between 0 and 1. After applying sigmoid function to the polarity score we obtain the output of the binary classificatin model.

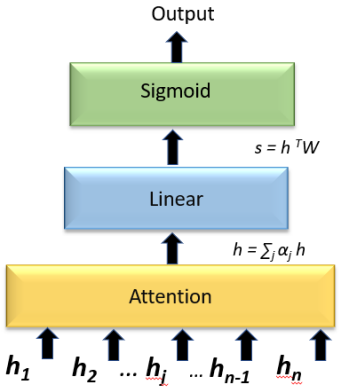


Figure 15: Classification Model with Attention Mechanism

3.2.3 Dataset

In this study, we have used two datasets, namely, Amazon cell phone reviews (ACR) and Amazon Kindle reviews (AKR). Both of these datasets are obtained from the Kaggle website and are publicly accessible.

- **ACR**: The ACR dataset comprises 82,815 reviews that were posted on the Amazon website for various cell phone models from the year 2004 to September 2019 [58].
- **AKR**: This dataset consists of a limited portion of product reviews from the Amazon Kindle Store category. It includes a 5-core subset, meaning that each entry in the dataset has at least 5 reviews from different reviewers for each product [59]. It contains 12000 review text.

Table 14: Hyperparameters used in the Proposed Model

Hyperparameter	**Value/Range**
Filter Sizes	[1,2,3,5,7]
Number of filters	36
Max Features	21000
Embedding Size	300
Kernel Initializer	glorot_uniform
Activation Function	elu
Attention Dense Units	36
Attention Activation	softmax
Dropout rate	0.3
Dense Layer Units (post-attention)	36
Dense Layer Activation (post-attention)	relu
Output Dense Layer Units	1
Output Activation	sigmoid
Dropout Rate	0.3
Optimizer	adam
Loss Function	binary_crossentropy

3.2.4 Data annotation

The study considers ratings of 1, 2, 4, and 5, excluding the neutral rating of 3. Ratings of 4 and 5 are labeled as positive (1), while ratings of 1 and 2 are labeled as negative (0). The data is split into 80% for training and 20% for testing, ensuring the model learns patterns effectively while maintaining a dedicated portion for evaluation to prevent overfitting.

3.2.5 Evaluation metrics

The evaluation of our study employs key performance metrics such as accuracy (A_c), precision (P_r), recall (R_e), and F1-score (F_s). To further support our analysis, we utilize a confusion matrix and bar plots, which visually represent the performance of the models. The confusion matrix helps in identifying true positives (T_{pos}), true negatives (T_{neg}), false positives (F_{pos}), and false negatives (F_{neg}), offering a clear picture of prediction accuracy. The bar plots complement this by providing an intuitive graphical comparison of model performance across the selected metrics. All evaluation metrics are determined through the application of equations 8 - 7.

$$A_c = \frac{T_{pos} + T_{neg}}{T_{pos} + T_{neg} + F_{pos} + F_{neg}} \tag{4}$$

$$R_e = \frac{T_{pos}}{T_{pos} + F_{neg}} \tag{5}$$

$$P_r = \frac{T_{pos}}{T_{pos} + F_{pos}} \tag{6}$$

$$F_s = 2 \cdot \frac{P_r \cdot R_e}{P_r + R_e} \tag{7}$$

3.2.6 Result analysis

Table 15: Experimental results of the proposed model with existing models on balanced dataset of ACR.

Model	Embedding	Accuracy	Precision	Recall	F-Score	AUC
LSTM	Glove	0.899	0.923	0.874	0.898	0.959
	Word2Vec	0.894	0.888	0.906	0.897	0.959
BiLSTM	Glove	0.897	0.887	0.911	0.898	0.956
	Word2Vec	0.892	0.886	0.899	0.893	0.954
GRU	Glove	0.895	0.891	0.902	0.896	0.958
	Word2Vec	0.884	0.895	0.871	0.883	0.950
Att+BiLSTM	Glove	0.894	0.909	0.872	0.890	0.960
	Word2Vec	0.896	0.914	0.870	0.892	0.960
1D CNN	Glove	0.892	0.926	0.853	0.888	0.957
	Word2Vec	0.894	0.905	0.881	0.893	0.955
2D CNN	Glove	0.909	0.916	0.903	0.910	0.962
	Word2Vec	0.909	0.914	0.905	0.910	0.963
Att+2D CNN	Glove	**0.919**	**0.922**	**0.918**	**0.920**	**0.972**
	Word2Vec	**0.915**	**0.928**	**0.901**	**0.915**	**0.971**

Table 16: Experimental results of the proposed model with existing models on unbalanced dataset of ACR.

Model	Embedding	Accuracy	Precision	Recall	F-Score	AUC
LSTM	Glove	0.910	0.942	0.930	0.936	0.963
	Word2Vec	0.913	0.934	0.944	0.939	0.962
BiLSTM	Glove	0.909	0.935	0.937	0.936	0.958
	Word2Vec	0.908	0.928	0.944	0.936	0.956
GRU	Glove	0.905	0.930	0.937	0.933	0.956
	Word2Vec	0.909	0.932	0.941	0.936	0.956
Att+BiLSTM	Glove	0.906	0.924	0.945	0.935	0.947
	Word2Vec	0.909	0.926	0.946	0.936	0.959
1D CNN	Glove	0.900	0.939	0.919	0.929	0.958
	Word2Vec	0.906	0.917	0.953	0.935	0.958
2D CNN	Glove	0.913	0.940	0.937	0.938	0.967
	Word2Vec	0.916	0.947	0.934	0.940	0.968
Att+2D CNN	Glove	**0.922**	**0.944**	**0.947**	**0.945**	**0.971**
	Word2Vec	**0.921**	**0.938**	**0.953**	**0.945**	**0.972**

3.2.6.1 Proposed Model Performance on Balanced and Unbalanced ACR Datasets

The experimental results in Tables II and III highlight the performance of various deep learning models on both balanced and unbalanced ACR datasets using GloVe and Word2Vec embeddings. Among them, the proposed Att+2D CNN model with GloVe achieved the highest accuracy (0.919), precision (0.922), recall (0.918), and F1-score (0.920). The same model with Word2Vec also showed strong performance. Models with GloVe consistently outperformed those with Word2Vec. Attention mechanisms significantly improved performance when added to BiLSTM and 2D CNN models, with Att+2D CNN outperforming all others. Additionally, 2D CNN performed better than 1D CNN, likely due to its better handling of spatial patterns. Overall, the proposed model with GloVe embeddings proved most effective.

In the case of the unbalanced ACR dataset, the proposed Att+2D CNN model with Word2Vec and Glove embeddings achieved the highest accuracy of 0.922 and 0.921 respectively. Comparatively, the Att+2D CNN model with GLoVe embeddings achieved a good AUC of 0.971. The Att+2D CNN and 2D CNN models demonstrated nearly identical accuracies, with only minor variations observed. When comparing the performance of the 1D CNN model with the 2D CNN model, the 1D CNN model showed slightly lower accuracy than the 2D CNN model. Among the BiLSTM, GRU, Att+BiLSTM, and 1D CNN models, they exhibited similar performance, except for the LSTM model, which showed some variation.

In figure 16 and 19 the first five epochs, the model shows significant improvement in both training loss and accuracy, indicating effective learning from the training data. Validation loss and accuracy also demonstrate positive trends, signifying promising generalization to new data. During epochs 6-15, there is a continued decrease in training loss and an increase in training accuracy, maintaining positive developments in validation metrics. In epochs 16-30, the model achieves a low training loss and high accuracy, but the increasing gap between training and validation curves suggests potential overfitting. The confusion matrices are displayed in figures 20 and 23.

In conclusion, based on the experimental results, the Att+2D CNN model with Glove embeddings consistently achieves the highest performance across all metrics in both balanced and unbalanced ACR datasets.

3.2.6.2 Proposed Model Performance on Balanced and Unbalanced AKR Datasets

The experimental results presented in Table IV and Table V provide insights into the performance of various existing deep learning models, including LSTM, BiLSTM, GRU, Att+BiLSTM, 1D CNN, 2D CNN and the proposed Att+2D CNN model. These models were evaluated on both balanced and unbalanced AKR datasets using Glove and Word2Vec embeddings. The key observations are as follows:

In the case of the balanced AKR dataset, the proposed Att+2D CNN model with Word2Vec embeddings achieved the highest accuracy of 0.903, the highest F1-score of 0.901, and an AUC of 0.964. The proposed Att+2D CNN model with Glove embeddings achieved accuracy of 0.894, an F1-score of 0.892, and an AUC of 0.961. The 2D CNN model with word2vec and Glove embeddings also demonstrated a good accuracy of 0.889. When com-

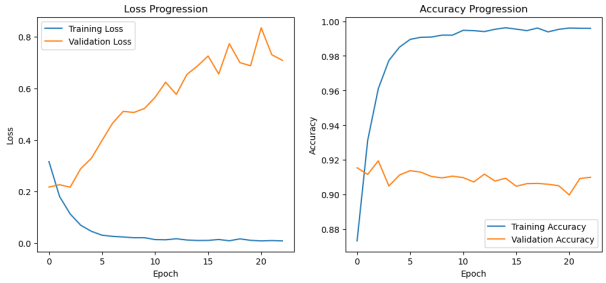


Figure 16: Loss Curve for the balanced ACR dataset using Glove embedding.

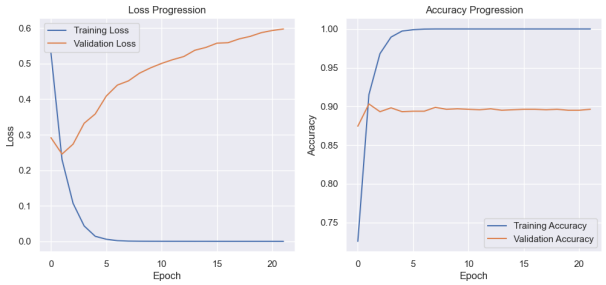


Figure 17: Loss Curve for the balanced AKR dataset using Word2Vec embedding.

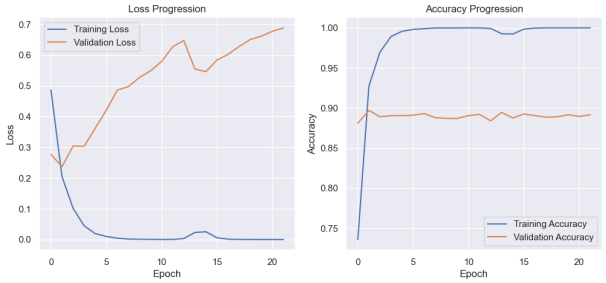


Figure 18: Loss Curve for the unbalanced AKR dataset using Word2Vec embedding.

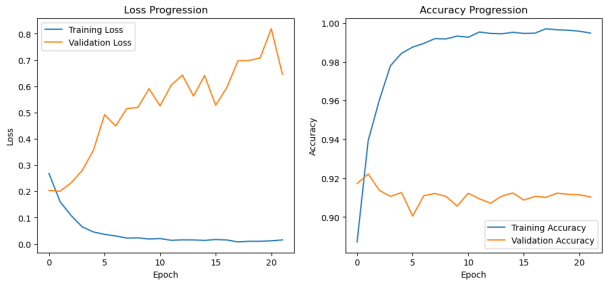


Figure 19: Loss Curve for the unbalanced ACR dataset using Glove embedding.

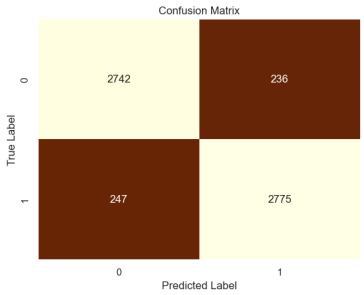


Figure 20: Confusion matrix for the balanced ACR dataset using Glove embedding.

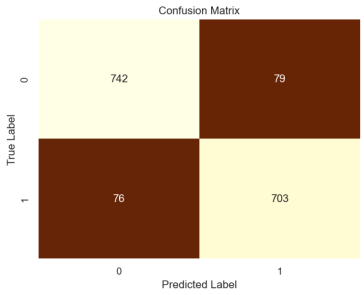


Figure 21: Confusion matrix for the balanced AKR dataset using Word2Vec embedding.

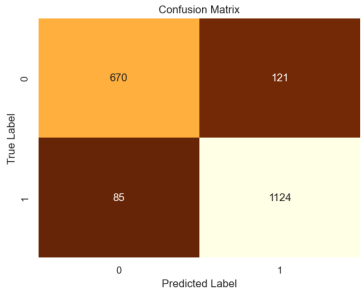


Figure 22: Confusion matrix for the unbalanced AKR dataset using Word2Vec embedding.

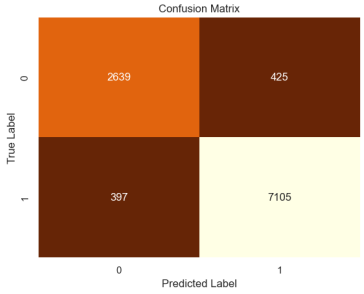


Figure 23: Confusion matrix for the unbalanced ACR dataset using Glove embedding.

Table 17: Experimental results of the proposed model with existing models on balanced dataset of AKR.

Model	Embedding	Accuracy	Precision	Recall	F-Score	AUC
LSTM	Glove	0.869	0.850	0.904	0.876	0.947
	Word2Vec	0.863	0.911	0.812	0.859	0.947
BiLSTM	Glove	0.864	0.852	0.882	0.867	0.935
	Word2Vec	0.866	0.892	0.833	0.862	0.929
GRU	Glove	0.872	0.894	0.846	0.869	0.933
	Word2Vec	0.844	0.827	0.871	0.848	0.925
Att+BiLSTM	Glove	0.856	0.837	0.881	0.859	0.937
	Word2Vec	0.869	0.868	0.868	0.868	0.943
1D CNN	Glove	0.869	0.873	0.866	0.869	0.947
	Word2Vec	0.881	0.893	0.868	0.880	0.950
2D CNN	Glove	0.889	0.879	0.903	0.891	0.954
	Word2Vec	0.889	0.881	0.895	0.888	0.954
Att+2D CNN	Glove	**0.894**	**0.889**	0.895	**0.892**	**0.961**
	Word2Vec	**0.903**	**0.899**	**0.902**	**0.901**	**0.964**

Table 18: Experimental results of the proposed model with existing models on unbalanced dataset of AKR.

Model	Embedding	Accuracy	Precision	Recall	F-Score	AUC
LSTM	Glove	0.863	0.894	0.876	0.885	0.939
	Word2Vec	0.875	0.879	0.921	0.899	0.942
BiLSTM	Glove	0.874	0.880	0.916	0.898	0.938
	Word2Vec	0.864	0.863	0.920	0.891	0.934
GRU	Glove	0.876	0.872	0.931	0.901	0.941
	Word2Vec	0.857	0.904	0.854	0.878	0.934
Att+BiLSTM	Glove	0.876	0.896	0.899	0.898	0.937
	Word2Vec	0.881	0.891	0.916	0.903	0.938
1D CNN	Glove	0.878	0.906	0.890	0.898	0.952
	Word2Vec	0.884	0.898	0.911	0.904	0.951
2D CNN	Glove	0.886	0.911	0.900	0.906	0.955
	Word2Vec	0.890	0.901	0.919	0.910	0.958
Att+2D CNN	Glove	**0.895**	**0.904**	**0.926**	**0.915**	**0.959**
	Word2Vec	**0.897**	**0.903**	**0.930**	**0.916**	**0.965**

paring the GRU model with Word2Vec embeddings, it exhibited a slightly lower accuracy of 0.844 and an AUC of 0.925. Among the BiLSTM models, the one incorporating attention mechanism outperformed the regular BiLSTM model, indicating that attention improved the results. The LSTM, 1D CNN and GRU models showed relatively similar results in the balanced AKR dataset.

In the case of the unbalanced AKR dataset, the proposed Att+2D CNN model with word2Vec embeddings achieved the highest accuracy of 0.897, with a precision of 0.903, recall of 0.930, F1-score of 0.916, and an AUC of 0.965. Similarly, the 2D CNN model with Word2Vec embeddings also demonstrated strong performance, with an accuracy of 0.890 and an AUC of 0.958. When compared to other model, the LSTM and BiLSTM models showed slightly lower accuracy and performance metrics. The GRU model, along with Att+BiLSTM and 1D CNN models, exhibited relatively similar results. In the unbalanced AKR dataset, Word2Vec embedding performed well.

In figures 17 and 18 the first five epochs, the model experiences a substantial reduction in both training loss and an increase in accuracy, indicating effective learning from the training data. Validation metrics also demonstrate improvement, suggesting positive generalization to new data. As the training progresses, the model achieves near-perfect accuracy, reaching 100%, while the validation loss remains relatively stable. However, the increasing validation loss after epoch 15 suggests a potential risk of overfitting. The confusion matrices are displayed in figures 21 and 22.

In conclusion, after considering the evaluation metrics and the choice of embeddings, the proposed Att+2D CNN model consistently outperforms other existing models in both balanced and unbalanced ACR and AKR datasets. For the ACR dataset, whether balanced or unbalanced, Glove embedding demonstrates notable accuracy, while for the AKR dataset, Word2Vec embedding performs well in both balanced and unbalanced scenarios. Moreover, models incorporating attention mechanisms, such as Att+BiLSTM, consistently exhibit improved results compared to their counterparts. The 1D CNN and GRU models demonstrate relatively similar performance. Overall, the choice of embeddings proves to be crucial, with Word2Vec consistently outshining Glove embeddings.

In summary, we can conclude that the two-dimensional CNN with an attention layer yields favorable results in our work compared to 2D CNN and 1D CNN. Additionally, it is evident that the 2D CNN model excels in handling both unbalanced and balanced datasets, showcasing its efficiency in text classification.

The performance measures for both Amazon datasets with respect to Glove and Word2Vec are presented in Figure 24 and 25. The bar plots clearly show that the 2D CNN with attention outperforms other models. Its performance metrics in terms of accuracy, precision, recall, and F1 score surpass those of other models.

3.3 Fine-Tuning XLNet for Amazon Review Sentiment Analysis: A Comparative Evaluation of Transformer Models

Large Language Models (LLMs) have significantly improved sentiment analysis (SA) through enhanced accuracy and efficiency. Transformers, introduced by Vaswani et al. [60], utilize self-attention and multi-head attention mechanisms to better capture long-range dependencies and contextual information.

Several transformer-based models—BERT, RoBERTa, DistilBERT, ALBERT, ELEC-

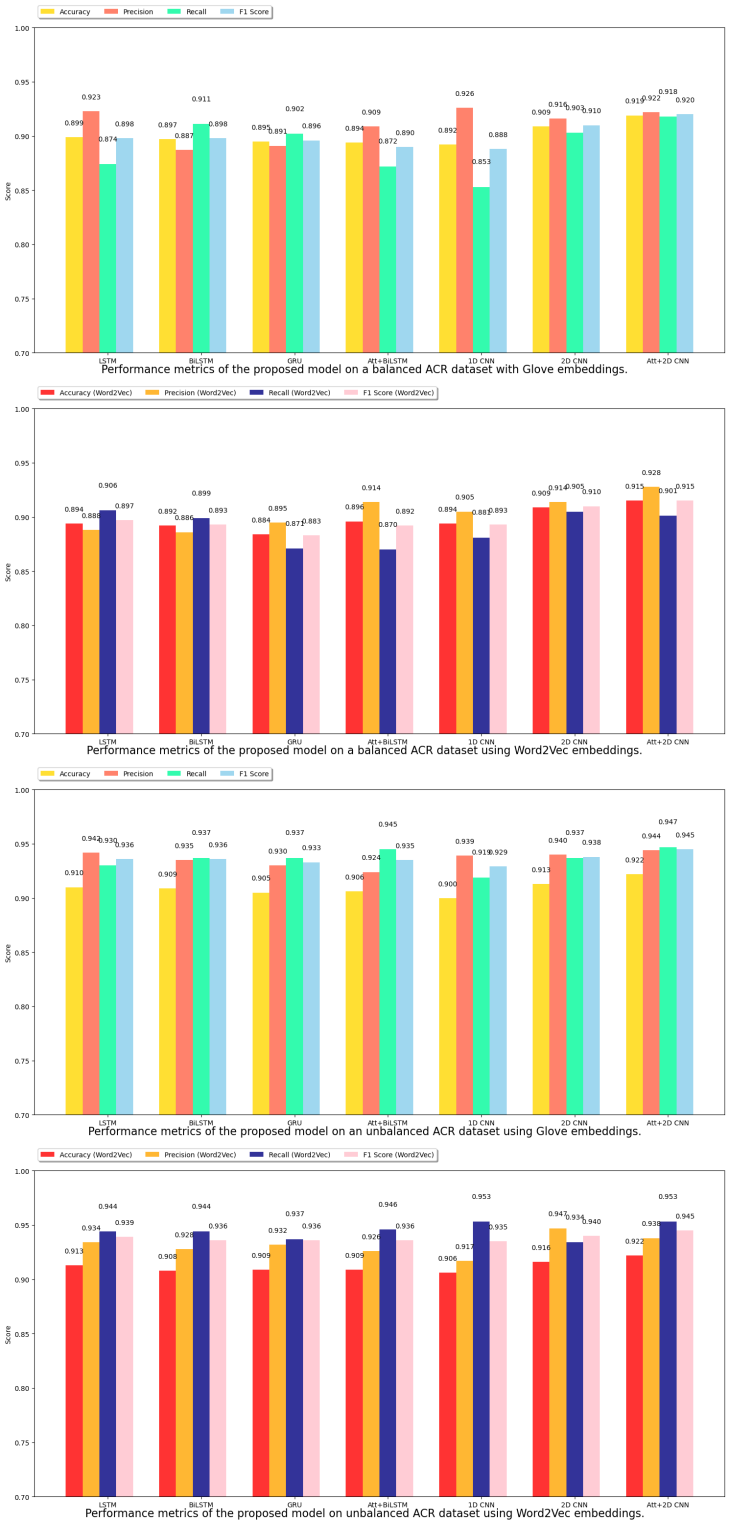

Accuracy
Precision
Recall
F1 Score
Score
LSTM
BiLSTM
GRU
Att+BiLSTM
1D CNN
2D CNN
Att+2D CNN
Performance metrics of the proposed model on a balanced ACR dataset with Glove embeddings.
Accuracy (Word2Vec)
Precision (Word2Vec)
Recall (Word2Vec)
F1 Score (Word2Vec)
Performance metrics of the proposed model on a balanced ACR dataset using Word2Vec embeddings.
Performance metrics of the proposed model on an unbalanced ACR dataset using Glove embeddings.
Performance metrics of the proposed model on unbalanced ACR dataset using Word2Vec embeddings.

Figure 24: The performance metrics of balanced and unbalanced ACR dataset using Glove and Word2vec embedding.

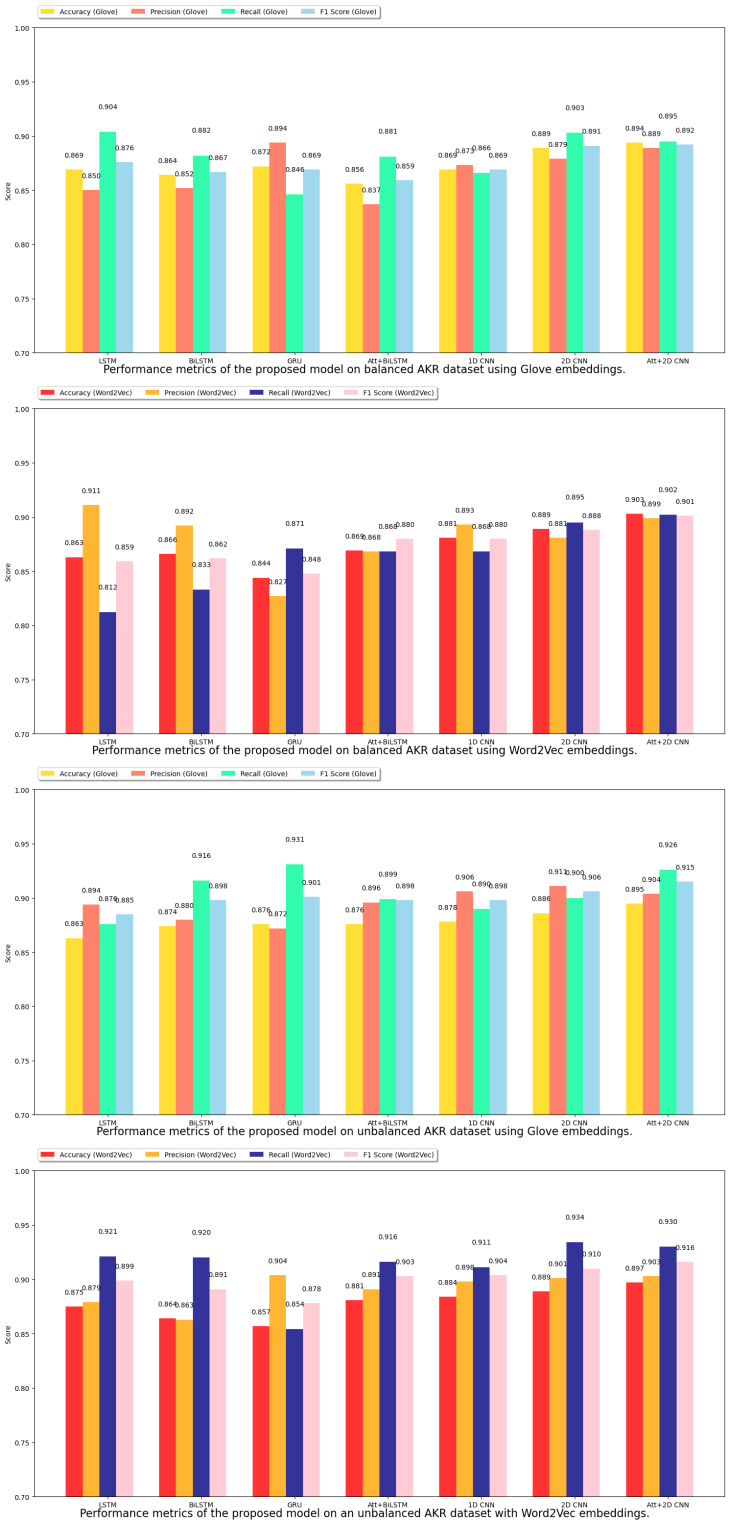

Accuracy (Glove)
Precision (Glove)
Recall (Glove)
F1 Score (Glove)
Score
LSTM
BiLSTM
GRU
Att+BiLSTM
1D CNN
2D CNN
Att+2D CNN
Performance metrics of the proposed model on balanced AKR dataset using Glove embeddings.
Accuracy (Word2Vec)
Precision (Word2Vec)
Recall (Word2Vec)
F1 Score (Word2Vec)
Performance metrics of the proposed model on balanced AKR dataset using Word2Vec embeddings.
Performance metrics of the proposed model on unbalanced AKR dataset using Glove embeddings.
Performance metrics of the proposed model on an unbalanced AKR dataset with Word2Vec embeddings.

Figure 25: The performance metrics of balanced and unbalanced AKR dataset using Glove and Word2vec embedding.

TRA, and XLNet—have been evaluated for SA tasks. BERT employs masked language modeling, whereas XLNet uses permutation language modeling for improved context handling [61]. ELECTRA adopts a unique approach by replacing tokens and training the model to detect inaccuracies.

Traditional models such as CNN, BiLSTM, BiGRU, and their hybrids (CNN-LSTM, CNN-BiLSTM), along with GloVe embeddings, were also explored. Experiments were conducted on two Amazon datasets—earphones and personal computers. XLNet consistently outperformed other models, followed by ELECTRA. The study further analyzed the effects of different batch sizes, optimizers (Adam, Nadam), and learning rates on model performance.

3.3.1 Framework Methodology

In the methodology section, data processing and training–test splitting procedures are described. In addition, an in-depth description of the proposed model is provided. The research workflow is illustrated in Fig. 26.

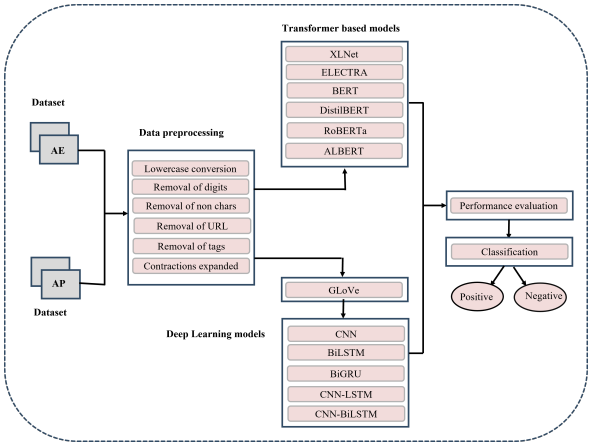


Figure 26: Flowchart of the adopted methodology.

3.3.1.1 Fine-tuned XLNet model

XLNet is a language model that is pre-trained and represents an enhanced version of BERT and GPT. It is known for its ability to capture effectively bidirectional context [61]. It uses transformer-XL [62] as its fundamental architecture for unsupervised learning. Unlike BERT, XLNet employs a permutation language modeling objective that enables it to consider all possible permutations of the input tokens during training. This approach enhances the model's understanding of context and improves its performance. The XLNet architecture is illustrated in Fig. 27.

The permutation-based language model described in Equation 8 works with an input sequence $[1, 2, 3, \ldots, T]$, where T represents the sequence length. The objective function can be written as

$$\max_{\theta} E_{z \sim S_T} \left[\sum_{t=1}^{T} \log p_\theta(z_t \mid z_{<t}) \right] \tag{8}$$

where S_T is the set of all possible permutations in a sequence of length T. For a specific permutation $z \in S_T$, z_t refers to the tth element and $z_{<t}$ represents the elements that precede z_t (i.e., the first $t-1$ tokens) in that permutation. The model uses the preceding tokens to predict the next.

To illustrate this, we considered the input sequences $\{\text{apple}, \text{banana}, \text{cherry}, \text{date}\}$. The potential permutation order for this sequence was $4! = 24$. Let $\{\text{banana} \rightarrow \text{apple} \rightarrow \text{date} \rightarrow \text{cherry}\}$, and $\{\text{cherry} \rightarrow \text{date} \rightarrow \text{banana} \rightarrow \text{apple}\}$ as two sample permutation orders. To predict *date*, the model considers *banana* and *apple* as context. The probability is computed as follows: $P(\text{date}|\text{banana}, \text{apple})$. For the second order, when predicting an *apple*, all preceding words were considered and expressed as $P(\text{apple}|\text{cherry}, \text{date}, \text{banana})$. This approach allows the model to learn the dependencies between tokens of various orders, enhancing its ability to capture complex relationships within a sequence.

The input layer of the XLNet-based text classification model was designed to process efficiently encoded text data. It primarily consists of a single input tensor named "input_ids." This tensor is of dtype "int32" and contains the tokenized and encoded text as integer IDs corresponding to the XLNet vocabulary. While only "input_ids" is explicitly defined in the input layer, the XLNet model internally also uses "attention_mask" and "token_type_ids." These additional inputs help the model focus on relevant parts of the text and distinguish among different sequences, although in this single-sequence classification task, "token_type_ids" are less crucial. This input layer design allows the model to handle variable-length text inputs efficiently with padding to ensure consistent dimensions across all samples in a batch.

After the input layer, the text data flow through XLNet's internal architecture. The encoded inputs are then processed by XLNet's transformer layers, which consist of multiple stacked self-attention and feedforward sublayers. These layers utilize a multihead attention mechanism to capture contextual information from the input tokens by computing the attention scores between every token pair in the sequence. XLNet's unique permutation-based training strategy ensures that each token can attend to any other token in the sequence regardless of its position, leading to a more comprehensive understanding of the relationships within the text.

The output from the transformer layers is a contextualized representation of the input sequence, referred to as the last hidden state. For text classification tasks, we typically require only a single vector representation for each sequence. XLNet provides a sequence summary layer that reduces the dimensionality of the last hidden state by extracting the relevant features. This summarized output is then passed on to the classification head of the model.

In this case, the classification head consists of a dense layer called `logits_proj` that maps the summarized output into a set of logits, each representing the score for one of the possible class labels. Because this is a binary classification task, the output consists of two logits corresponding to the two labels (e.g., positive vs. negative sentiment). These logits are then passed to a softmax function, which converts them into probabilities that sum to one, allowing us to interpret the model's predictions as the likelihood of each class.

To train the model, the compiled model uses sparse categorical cross-entropy as the loss function. The Adam optimizer was employed to adjust the model's parameters, and sparse categorical accuracy was used as the evaluation metric to quantify the performance of the model during training and validation. As the training progressed, two important

callbacks were integrated into the workflow, namely EarlyStopping and ModelCheckpoint. EarlyStopping ensures that training stops if the validation loss does not improve for a set number of epochs, thereby preventing the model from overfitting. ModelCheckpoint saves the model's weights whenever there is an improvement in the validation loss, ensuring that the best-performing model is retained at the end of training.

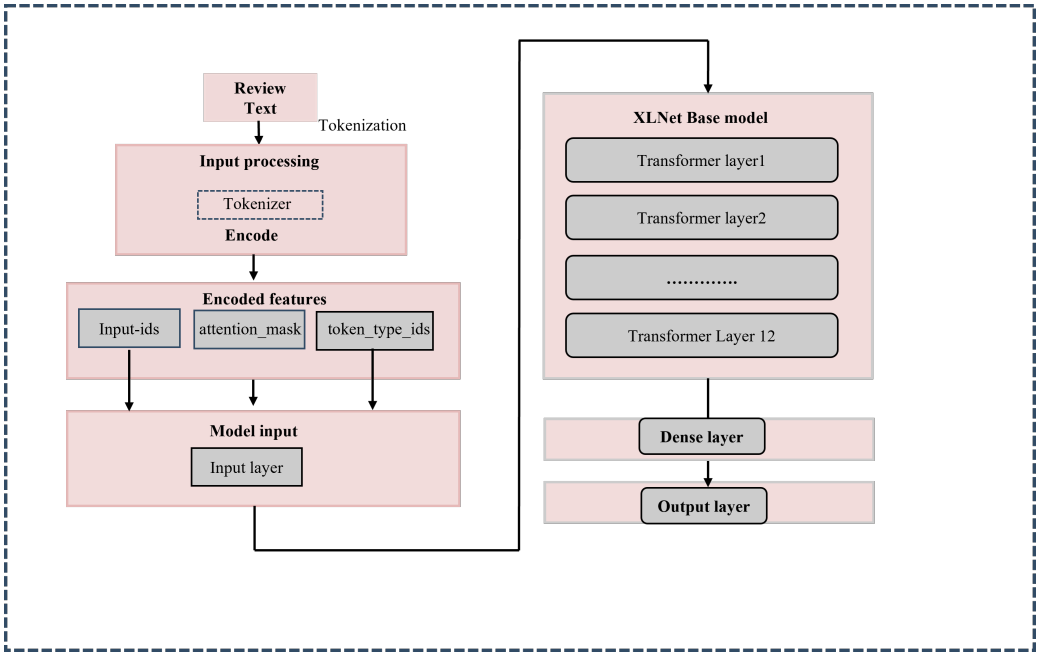


Figure 27: XLNet architecture.

The hyperparameters used in the model are listed in Table 19. Pseudocode for Amazon review SA using fine-tuned XLNet, BERT, DistilBERT, RoBERTa, ELECTRA, and ALBERT are presented in the algorithms 1.

3.3.2 Result analysis

3.3.2.1 Performance comparison of DL models on AE dataset

In Table 20, the top three models—1D CNN, BiLSTM, and BiGRU—exhibit nearly identical overall performances, achieving accuracy and F1 scores close to 91.5%. However, the CNN-LSTM and CNN-BiLSTM models show a slight decrease in performance, with their metrics trailing by approximately three percentage points compared with the highest-performing models.

The 1D CNN achieved the highest accuracy of 91.7% and a precision of 92.9%. The BiLSTM model excelled in recall with a score of 92.0%, indicating its effectiveness in identifying all relevant instances. BiGRU demonstrated a well-balanced performance across all metrics, with scores consistently above 91%. In contrast, the CNN-LSTM and CNN-BiLSTM models achieved respectable scores above 88% for all metrics. This suggests that simpler models, such as 1D CNN, BiLSTM, and BiGRU outperform more complex hybrid models, such as CNN-LSTM and CNN-BiLSTM when applied on the AE dataset.

Component	Description
pretrained model	`xlnet-base-cased`, base XLNet trained on cased text
tokenizer	initializes XLNet tokenizer for input text
maximum length	maximum sequence length: 100 tokens, truncated/padded
batch size	training batch size: 8 for efficiency
attention heads	XLNet employs 12 multihead attention mechanisms
number of labels	binary classification: 2 labels
optimizer	Adam optimizer, lr: 2e-5
loss function	sparse categorical cross-entropy
metrics	sparse categorical accuracy
number of epochs	trained for five epochs
total parameters	117310466 parameters
trainable parameters	all parameters trainable
sequence summary layer	condenses sequence representations into a single summary vector of size 768
logits projection layer	maps summary vector to final output logits, indicating the probabilities for each class
early stopping	monitors "val_loss," patience: three epochs
checkpointing	saves model based on "val_loss," saves best only

Table 19: Hyperparameter values for the fine-tuned XLNet model.

Model	Metrics(%)			
	Accuracy	Precision	Recall	F1 score
1D CNN	91.7	92.9	90.2	91.5
BiLSTM	91.5	90.9	92.0	91.4
BiGRU	91.6	91.8	91.2	91.5
CNN-LSTM	88.7	89.3	87.8	88.5
CNN-BiLSTM	88.7	88.5	88.8	88.7

Table 20: Comparative analysis of model performance on the AE dataset.

3.3.2.2 Comparative analysis of transfer learning models on the AE dataset

As shown in Table 21, XLNet and ELECTRA achieved the highest accuracies of 95.2% and 95.0%, respectively, while the lowest accuracy was recorded by RoBERTa at 91.2%. XLNet also achieved the highest F1 score, striking a balance between precision and recall, closely followed by ELECTRA with an F1 score of 94.9%. In contrast, RoBERTa had the lowest F1 score (90.9%). In terms of EMR, ELECTRA and XLNet tied at 0.95, meaning that 95% of the predictions were correct for all classes. For the Hamming loss, XLNet and ELECTRA demonstrated the best performances with 4.79% and 4.96%, respectively. Among the weighted metrics, XLNet and ELECTRA exhibited the highest weighted scores across precision, recall, and F1. However, XLNet has the longest training time compared with the other models, while DistilBERT has the shortest training time.

From these observations, we conclude that XLNet has the highest values for accuracy,

Algorithm 1 Pseudocode for review sentiment analysis and classification for transformers

Input: Datasets R_1 and R_2
Output: Classification Results and Performance Metrics for R_1 and R_2
for each dataset R in $\{R_1, R_2\}$ **do**
    Step 1: Data Preprocessing
    $R_P \leftarrow \texttt{Preprocess\_Reviews}(R)$
    / Expand contractions, remove URLs, HTML tags, digits, convert to lowercase, and remove non-alphabetic characters */*
    Step 2: Tokenization and Vectorization
    $R_T \leftarrow \texttt{Tokenize\_And\_Vectorize}(R_P)$
    / Convert preprocessed text into tokenized vectors for Transformer-based models */*
    Step 3: Training–Testing Splitting
    $R_{train}, R_{test} \leftarrow \texttt{TrainTestSplit}(R_T, 0.3)$
    $R_{train}, R_{val} \leftarrow \texttt{TrainValidationSplit}(R_{train}, 0.2)$
    / Split the dataset into training, testing, and validation subsets */*
    for each model M in {Fine-tuned XLNet, BERT, DistilBERT, ALBERT, RoBERTa, ELECTRA} **do**
        for $i = 1$ to n **do**
            Step 4: Process Each Review from the Test Set
            $Review_i \leftarrow \texttt{Select\_Review}(R_{test}, i)$
            $Result_i \leftarrow \texttt{ApplyModel}(M, Review_i)$
            / Apply the selected model for sentiment classification */*
        end for
        Step 5: Compute Performance Metrics
        $Acc_M \leftarrow \texttt{ComputeAccuracy}(R_{test}, Results_M)$
        $Prec_M \leftarrow \texttt{ComputePrecision}(R_{test}, Results_M)$
        $Rec_M \leftarrow \texttt{ComputeRecall}(R_{test}, Results_M)$
        $F1_M \leftarrow \texttt{ComputeF1Score}(R_{test}, Results_M)$
        $EMR_M \leftarrow \texttt{ComputeExactMatchRatio}(R_{test}, Results_M)$
        $HLoss_M \leftarrow \texttt{ComputeHammingLoss}(R_{test}, Results_M)$
        $WPrec_M \leftarrow \texttt{ComputeWeightedPrecision}(R_{test}, Results_M)$
        $WRec_M \leftarrow \texttt{ComputeWeightedRecall}(R_{test}, Results_M)$
        $WF1_M \leftarrow \texttt{ComputeWeightedF1Score}(R_{test}, Results_M)$
        / Compute and store various performance metrics for the model M */*
    end for
end for
return *Results* and *Metrics*
/ Return the classification results and performance metrics */*

precision, recall, F1 score, and the lowest Hamming loss; however, it is associated with the drawback of a longer training time. By contrast, DistilBERT is more efficient, with a much shorter training time and competitive performance, especially for less resource-intensive applications.

Metrics (%)	Distil BERT	ALBERT	RoBERTa	BERT	ELECTRA	XLNet
accuracy	92.0	93.7	91.2	93.0	95.0	**95.2**
precision	93.5	93.0	92.4	98.3	97.6	**95.8**
recall	90.2	94.4	89.5	87.4	92.3	**94.4**
F1 score	91.8	93.7	90.9	92.5	94.9	**95.1**
exact match ratio (EMR)	0.92	0.94	0.91	0.93	0.95	**0.95**
Hamming loss	8.00	6.33	8.83	7.00	4.96	**4.79**
weighted precision	92.0	93.6	91.2	93.5	95.1	**95.2**
weighted recall	92.0	93.6	91.1	93.0	95.0	**95.2**
weighted F1 score	92.0	93.6	91.1	92.9	95.0	**95.2**
training time	50:43	1:08:36	2:55:0	2:48:19	3:02:32	**8:30:14**

Table 21: Performance metrics and training efficiency on the AE dataset.

3.3.2.3 Performance comparison of DL models on the AP dataset

As shown in Table 22, the BiLSTM model achieved the highest accuracy of 91.9% and yielded an F1 score of 91.7%. The CNN-BiLSTM model also performed well with an accuracy of 91.5% and an F1 score of 91.3%. The BiGRU model offered a well-balanced performance with an accuracy of 90.5% and an F1 score of 90.3%, making it competitive with BiLSTM and CNN-BiLSTM. Among the DL models, CNN-LSTM and 1D CNN exhibited the lowest performance. Specifically, the 1D CNN recorded the lowest accuracy of 88.1% and yielded an F1 score of 87.2%. In summary, the BiLSTM and CNN-BiLSTM models are the top performers in terms of accuracy and F1 score, whereas the CNN-LSTM model excels in recall but falls short in precision. The 1D CNN lags behind all other models in performance metrics.

Model	Metrics(%)			
	Accuracy	Precision	Recall	F1 Score
1D CNN	88.1	93.5	81.7	87.2
BiLSTM	91.9	92.9	90.7	91.7
BiGRU	90.5	91.0	89.7	90.3
CNN-LSTM	89.8	86.4	94.3	90.2
CNN-BiLSTM	91.5	92.5	90.1	91.3

Table 22: Comparative analysis of model performance on the AP dataset.

3.3.2.4 Comparative analysis of transfer learning models on the AP dataset

In the comparative analysis of the transfer learning models on the AP dataset as shown in Table 23, XLNet achieved the highest accuracy at 95.0%, followed closely by ELECTRA (94.5%) and BERT (94.3%), whereas ALBERT recorded the lowest accuracy at 92.0%. For

the F1 score, XLNet and ELECTRA were both tied at 94.6%, showing a balance between precision and recall, whereas ALBERT lagged in order, with an F1 score of 92.4%. In terms of the EMR, BERT, ELECTRA, and XLNet all tied at 0.94, meaning that 94% of the predictions were correct across all classes. For the Hamming loss, ELECTRA and XLNet tied with the lowest values at 5.54%, whereas ALBERT recorded the highest Hamming loss at 7.96%, indicating more incorrect labels. Among the weighted metrics (precision, recall, F1), BERT, ELECTRA, and XLNet consistently performed the best with high scores. However, XLNet again had the longest training time at 7 h, 31 min, and 57 s, whereas DistilBERT was the fastest (only required 1 h, 3 min, and 34 s).

Metrics (%)	Distil BERT	ALBERT	RoBERTa	BERT	ELECTRA	XLNet
accuracy	93.3	94.3	92.0	94.3	94.5	**95.0**
precision	91.2	96.4	87.5	96.4	91.8	**91.8**
recall	95.7	92.0	97.9	92.0	97.6	**97.6**
F1 score	93.4	94.1	92.4	94.1	94.6	**94.6**
EMR	0.93	0.94	0.92	0.93	0.94	**0.94**
Hamming loss	6.71	5.67	7.96	7.50	5.54	**5.54**
weighted precision	93.4	94.4	92.6	92.5	94.6	**94.6**
weighted recall	93.2	94.3	92.0	92.5	94.6	**94.4**
weighted F1 score	93.2	94.3	92.0	92.5	94.4	**94.4**
training time	1:03:34	2:43:35	3:20:38	3:29:44	3:15:58	**7:31:57**

Table 23: Evaluation metrics comparison for transformer models on the AP dataset.

We performed a comprehensive evaluation of various DL and transfer learning models on the AE and AP datasets. Our results showed that simpler DL models, such as 1D CNN, BiLSTM, and BiGRU, performed comparably, achieving high accuracy and balanced metrics across both datasets. These models outperformed more complex hybrid models such as CNN-LSTM and CNN-BiLSTM, which exhibited a slight drop in performance metrics.

Among the transfer learning models, XLNet and ELECTRA consistently ranked as the top performers, achieving the highest scores for accuracy, precision, recall, F1 score, and Hamming loss across both datasets. However, XLNet's considerable training time remains a critical drawback, highlighting the tradeoff between performance and computational efficiency. DistilBERT stands out for its training efficiency, although it has a slightly lower performance, making it a viable option for applications with limited computational resources.

The confusion matrices of XLNet model for the AE and AP dataset are presented in Figs. 28 and 29, respectively. The XLNet model was fine-tuned using various learning rates, and the best performance was achieved at a learning rate of 2e-5, as listed in Table 25.

From the performance metrics presented in Tables 24 (AP dataset) and 26 (AE dataset), it can be inferred that the Nadam optimizer consistently outperformed the Adam optimizer in the AE dataset, particularly with a batch size of 16. In the AP dataset, the highest accuracy of 95.5% was achieved with the Nadam optimizer and a batch size of 16, whereas the AE dataset exhibited a peak accuracy of 94.6% for the same batch size. The Nadam

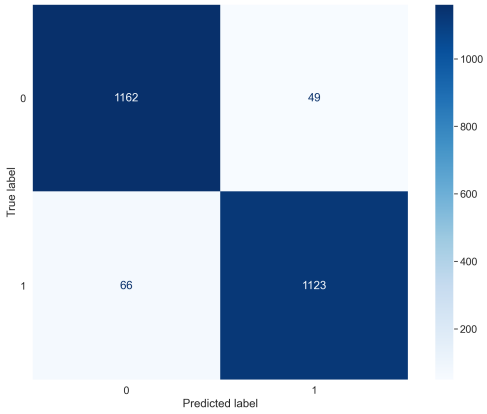


Figure 28: Confusion matrix of XL-Net on the AE dataset.

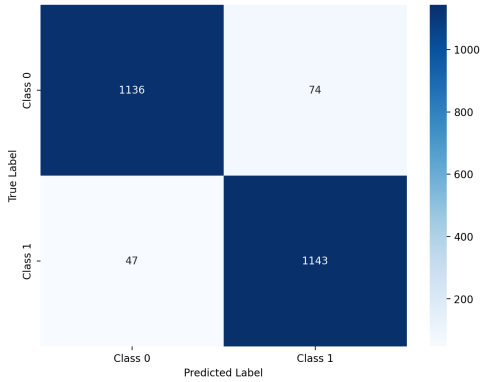


Figure 29: Confusion matrix of XL-Net on the AP dataset.

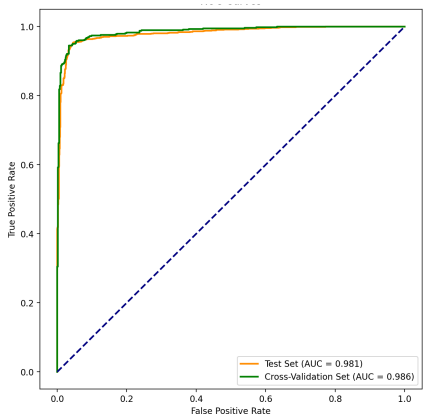


Figure 30: ROC curve for XL-Net based on the AE dataset.

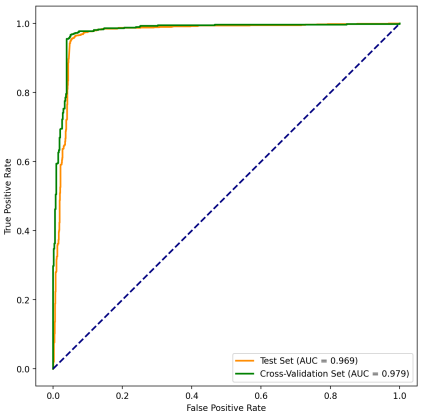


Figure 31: ROC curve for XL-Net based on the AP dataset.

Optimizer	Batch size	Accuracy	Precision	Recall	F1 score
Nadam	4	94.3	92.5	96.9	94.6
Nadam	8	94.7	95.9	93.4	94.6
Nadam	16	**95.5**	94.3	96.6	95.5
Adam	4	95.5	95.0	95.9	95.4
Adam	8	95.0	93.9	96.1	95.0
Adam	16	**95.7**	95.0	96.5	95.7

Table 24: Performance metrics for AP dataset with different batch sizes and optimizers.

optimizer also yielded superior precision and F1 scores, further confirming its effectiveness. For both datasets, batch sizes of 4, 8, and 16 emerged as favorable choices, with the Nadam optimizer demonstrating advantages in balancing accuracy, precision, and recall. This analysis highlights the critical role of the optimizer selection and batch size in enhancing the performance of the XLNet model in SA tasks.

3.4 Enhancing Sentiment Analysis of Amazon Reviews with Deep Hybrid Parallel Fusion of CNN and LSTM Using Pretrained Word Embeddings

This study addresses the limitations of traditional text representation methods like BoW and TF-IDF in sentiment analysis (SA) by leveraging pre-trained word embeddings such as

Learning rate	AE accuracy (%)	AP accuracy (%)
1e-05	94.3	94.8
2e-05	**95.2**	**95.0**
3e-05	93.5	94.5
4e-05	94.6	94.4
5e-05	94.1	93.7

Table 25: **Learning rate comparison for the accuracy of the AE and AP datasets.**

Optimizer	Batch size	Accuracy	Precision	Recall	F1 score
Nadam	4	93.3	92.0	94.8	93.4
Nadam	8	93.9	95.6	91.9	93.7
Nadam	16	94.6	97.1	91.9	94.4
Adam	4	88.9	86.7	91.8	89.1
Adam	8	95.2	95.8	94.4	95.1
Adam	16	93.5	94.5	92.3	93.4

Table 26: Performance metrics for the AE dataset with different batch sizes and optimizers.

GloVe, FastText, and Word2Vec, which capture richer semantic and syntactic information. To enhance performance, a novel Hybrid Parallel Fusion CNN-LSTM (HPFCL) model is proposed, which processes word vectors through parallel CNN and LSTM layers. The fused features are then passed to a meta-learner for sentiment prediction. Experimental results show that the HPFCL model outperforms existing approaches, demonstrating its effectiveness for SA on Amazon reviews.

we detail the methodology used to classify sentiment as positive or negative in Amazon reviews. The process commences with data preprocessing to ensure the data is prepared for analysis. Subsequently, pretrained word embeddings such as GloVe, FastText, and Word2Vec are employed. These embeddings are then fed into the Deep Hybrid Parallel Fused CNN-LSTM (HPFCL) model, which operates as the base learner. The CNN and LSTM components extract features from the data, which are then fused using a specific strategy. Finally, in the meta-learner phase, these fused features are inputted into a Gradient Boosting machine learning model for accurate sentiment classification. The proposed methodology is presented in Fig.32.

3.4.1 Proposed model (HPFCL)

HPFCL integrates two neural networks, leveraging their combined strengths to enhance classification tasks. The term "parallel" signifies that multiple models operate concurrently, utilizing parallel computing to boost efficiency during training and inference. Additionally, a fusion strategy is employed to extract and utilize features from both CNN and LSTM models. This method consists of two phases: the first phase acts as the base learner, while the second serves as the meta-learner. In the base learner, features are extracted in parallel from both CNN and LSTM models using word embedding features

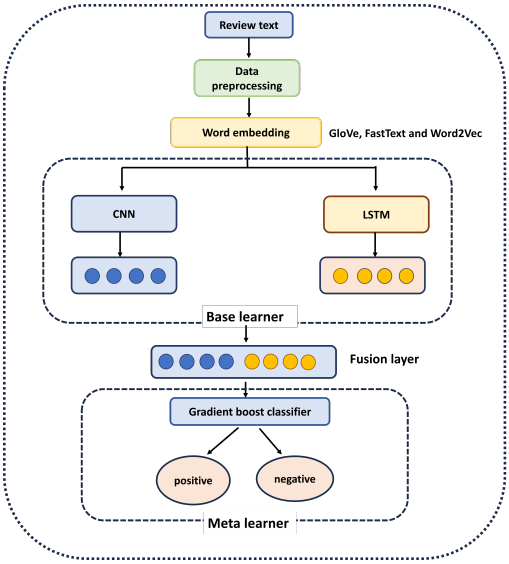


Figure 32: Workflow Diagram of the Study

as input. These features are then combined through the fusion strategy and passed to the Gradient Boosting model in the second phase to classify the sentiment of a review as positive or negative. The architecture of the proposed model is shown in Fig. 33.

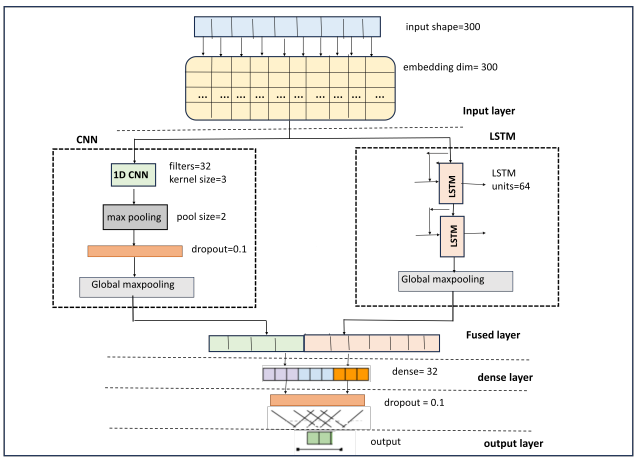


Figure 33: Architecture of the base learner

3.4.1.1 CNN-based feature extraction

By performing spatial filtering, convolutional layers extract meaningful patterns and features from the input. The CNN model incorporates convolution, pooling, and flattening operations to achieve this.

3.4.1.2 LSTM-based feature extraction

The LSTM layers sequentially process the embedded text data, considering the context of each word in the sentence by leveraging information from preceding words. The input sequence traverses the LSTM layer, which processes the sequence in its original order.

The output layer employs the final hidden state, or cell state, of the LSTM as a fixed-size representation of the entire input sequence.

3.4.1.3 Hybrid Fusion of CNN+LSTM

By combining the strengths of CNNs and LSTMs, the fusion mechanism enables the model to effectively utilize both spatial and temporal information, resulting in a more robust representation of the input data. In the proposed architecture, fusion is achieved through the concatenation of the two feature sets.

3.4.1.4 Classification

The sentiment of the input sequence is determined by the classification layer, which processes the fused features. In our hybrid model, we utilize a gradient-boosting classifier to predict whether an Amazon review is positive or negative. By incorporating this parallel fusion technique into the classification process, our model can leverage both local pattern recognition (from the CNN) and long-range context capture (from the LSTM). This approach has the potential to enhance the overall performance of sentiment classification tasks that involve sequential data. The hyperparameters used in base learner and meta learner are shown in Table 27 and Table 28.

Hyperparameter	**Value**
Input Shape	300
Embedding Dimension	300
Vocabulary Size	20000
LSTM Units	64
LSTM Return Sequences	True
CNN Filters	32
CNN Kernel Size	3
CNN Activation	ReLU
Max Pooling Size	2
Dropout Rate (CNN Branch)	0.1
Dropout Rate (Fusion Dense Layer)	0.1
Dense Units (Fusion Layer)	32
Output Activation	Sigmoid
Optimizer	Adam
Learning Rate	0.001
Loss Function	Binary Crossentropy
Metrics	Accuracy
Early Stopping Patience	3
Early Stopping Monitor	Validation Loss

Table 27: Hyperparameters used in the HPFCL model **(base learner)**

Hyperparameter	Value
n_estimators	100
random_state	42

Table 28: Hyperparameters used in the GradientBoost classifier **(meta learner)**

3.4.2 Results and analysis

Tables 29 and 30 show the experimental details about the performance of various models using GloVe, Word2Vec, and FastText embeddings.

3.4.2.1 Experimental analysis on GloVe

In the AMI dataset, we observe that the proposed model achieves highest F-score of 0.945, and accuracy of of 0.900. The BiLSTM and MC CNN-BiLSTM models achieved an identical F1 score of 0.937, but slightly lower accuracy scores compared to the proposed model.

In the AWC dataset, the proposed model achieved an accuracy of 0.886, F-score of 0.929, and AUC of 0.915. The LSTM and BiLSTM models achieved slightly lower accuracy scores, specifically 0.877 and 0.871, respectively. The aggregation performance in both datasets shows an improvement of approximately 2% with the use of GloVe.

Model	Embedding	Accuracy	Precision	Recall	F-Score	AUC
LSTM	Glove	0.879	0.880	0.999	0.936	0.727
BiLSTM	Glove	0.886	0.915	0.960	0.937	0.811
1D CNN	Glove	0.878	0.916	0.948	0.932	0.829
MC CNN-LSTM	Glove	0.879	0.899	0.971	0.934	0.813
MC CNN-BiLSTM	Glove	0.881	0.881	1.000	0.937	0.867
Proposed Model	**GloVe**	**0.900**	0.923	0.968	**0.945**	**0.875**
LSTM	FastText	0.882	0.883	0.998	0.937	0.780
BiLSTM	FastText	0.880	0.881	0.999	0.936	0.785
1D CNN	FastText	0.890	0.920	0.959	0.939	0.841
MC CNN-LSTM	FastText	0.887	0.889	0.997	0.940	0.811
MC CNN-BiLSTM	FastText	0.888	0.889	0.997	0.940	0.835
Proposed Model	**FastText**	**0.901**	0.919	0.973	**0.945**	**0.870**
LSTM	Word2Vec	0.880	0.904	0.965	0.934	0.772
BiLSTM	Word2Vec	0.851	0.916	0.915	0.915	0.815
1D CNN	Word2Vec	0.885	0.927	0.944	0.935	0.841
MC CNN-LSTM	Word2Vec	0.888	0.899	0.983	0.939	0.783
MC CNN-BiLSTM	Word2Vec	0.879	0.880	0.999	0.936	0.808
Proposed Model	**Word2Vec**	**0.900**	0.923	0.966	**0.944**	**0.867**

Table 29: Performance comparison of all models on unbalanced AMI dataset with GloVe, FastText, and Word2Vec embeddings

3.4.2.2 Experimental analysis on FastText

In the AMI dataset, the proposed model using FastText embeddings achieved F-score of 0.945, an accuracy of 0.901, and AUC of 0.870. 1D CNN and MC CNN-BiLSTM models

Model	Embedding	Accuracy	Precision	Recall	F-Score	AUC
LSTM	Glove	0.877	0.890	0.960	0.924	0.899
BiLSTM	Glove	0.871	0.885	0.959	0.920	0.895
1D CNN	Glove	0.866	0.890	0.944	0.916	0.890
MC CNN-LSTM	Glove	0.861	0.884	0.945	0.913	0.890
MC CNN-BiLSTM	Glove	0.869	0.878	0.965	0.919	0.900
Proposed Model	**Glove**	**0.886**	0.903	0.956	**0.929**	**0.915**
LSTM	FastText	0.873	0.895	0.948	0.921	0.901
BiLSTM	FastText	0.877	0.910	0.933	0.921	0.908
1D CNN	FastText	0.858	0.907	0.911	0.909	0.891
MC CNN-LSTM	FastText	0.866	0.893	0.940	0.916	0.897
MC CNN-BiLSTM	FastText	0.866	0.891	0.942	0.916	0.894
Proposed Model	**FastText**	**0.881**	**0.900**	0.952	**0.926**	**0.913**
LSTM	Word2Vec	0.872	0.894	0.946	0.920	0.896
BiLSTM	Word2Vec	0.876	0.881	0.971	0.924	0.895
1D CNN	Word2Vec	0.861	0.892	0.933	0.912	0.881
MC CNN-LSTM	Word2Vec	0.853	0.914	0.895	0.905	0.892
MC CNN-BiLSTM	Word2Vec	0.877	0.898	0.948	0.923	0.902
Proposed Model	**Word2Vec**	**0.881**	0.901	0.952	**0.926**	**0.917**

Table 30: Performance comparison of all models on unbalanced AWC dataset with GloVe, FastText, and Word2Vec embeddings

also performed well, with accuracy scores of 0.890 and 0.888, respectively. The 1D CNN model demonstrated high precision 0.920 and AUC 0.841.

The LSTM and MC CNN-LSTM models achieved accuracy scores of 0.882 and 0.887, respectively. 1D CNN and MC CNN-BiLSTM models also performed well, with accuracy scores of 0.890 and 0.888, respectively. The 1D CNN model demonstrated high precision (0.920) and AUC (0.841). The LSTM and MC CNN-LSTM models achieved accuracy scores of 0.882 and 0.887, respectively.

In the AWC dataset, the proposed model using FastText embeddings achieved an accuracy of 0.881, F-score of 0.926, and AUC of 0.913. The BiLSTM model also showed competitive performance with an accuracy of 0.877 and F1-score of 0.921. The 1D CNN model had the lowest accuracy compared to other models.

In both datasets, we observe that the proposed model using FastText embeddings generally outperforms other models in terms of accuracy, precision, recall, F-score, and AUC. The aggregation performance in both datasets shows an improvement of approximately 2% with the use of FastText.

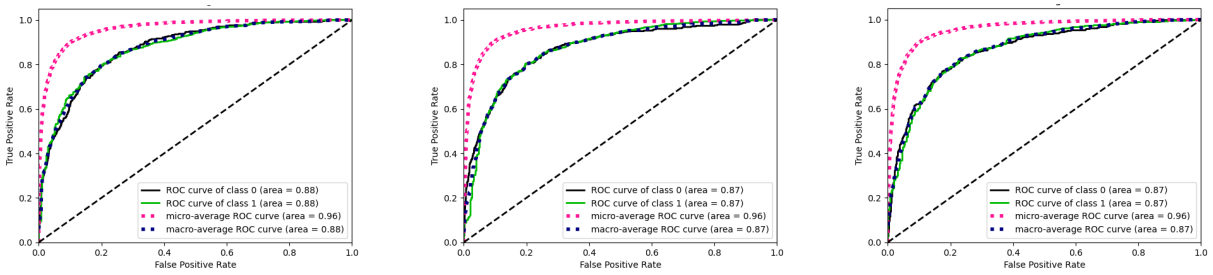
3.4.2.3 Experimental analysis on Word2Vec

In the AMI dataset, the performance of various models utilizing Word2Vec embeddings was evaluated. The proposed model utilizing Word2Vec embeddings achieved notable performance metrics, including an accuracy of 0.900, F-score of 0.944, and an AUC of 0.867.

Among the models, the MC CNN-LSTM model exhibited competitive performance with an accuracy of 0.888 and F-score of 0.939. The BiLSTM model achieved a lowest recall of 0.915. Conversely, the MC CNN-BiLSTM model displayed high recall at 0.999.

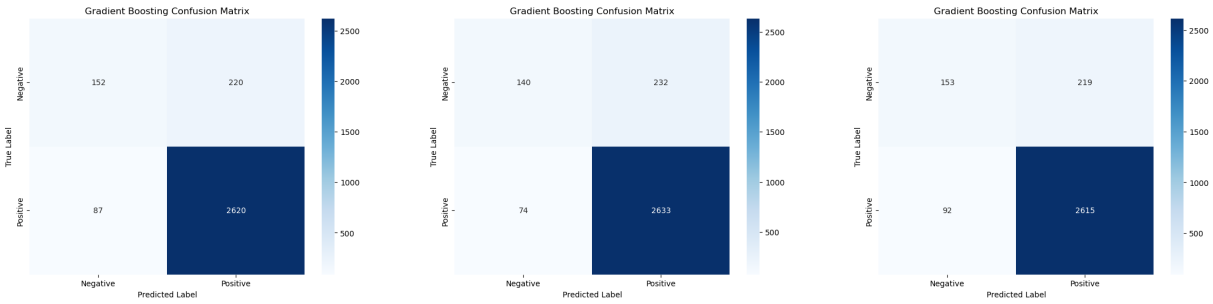
Similarly, in the AWC dataset, the proposed model utilizing Word2Vec embeddings demonstrated robust performance, yielding an accuracy of 0.881, F-score of 0.926, and AUC of 0.917.

Comparatively, the BiLSTM model in the AWC dataset showcased competitive performance, achieving an accuracy of 0.876 and F-score of 0.924. The LSTM model attained an accuracy of 0.872 and precision of 0.894, although it displayed a relatively lower recall of 0.946. Furthermore, the 1D CNN and MC CNN-BiLSTM models also demonstrated competitive performance with accuracies of 0.861 and 0.877, respectively.The ROC and confusion matrices for GloVe, Word2Vec, and FastText of the proposed model on the AMI dataset are shown in Fig. 34 and 35 and for the AWC dataset are shown in Figs.36 and 37.


True Positive Rate
False Positive Rate
ROC curve of class 0 (area = 0.88)
ROC curve of class 1 (area = 0.88)
micro-average ROC curve (area = 0.96)
macro-average ROC curve (area = 0.88)
ROC curve of class 0 (area = 0.87)
ROC curve of class 1 (area = 0.87)
micro-average ROC curve (area = 0.96)
macro-average ROC curve (area = 0.87)
ROC curve of class 0 (area = 0.87)
ROC curve of class 1 (area = 0.87)
micro-average ROC curve (area = 0.96)
macro-average ROC curve (area = 0.87)

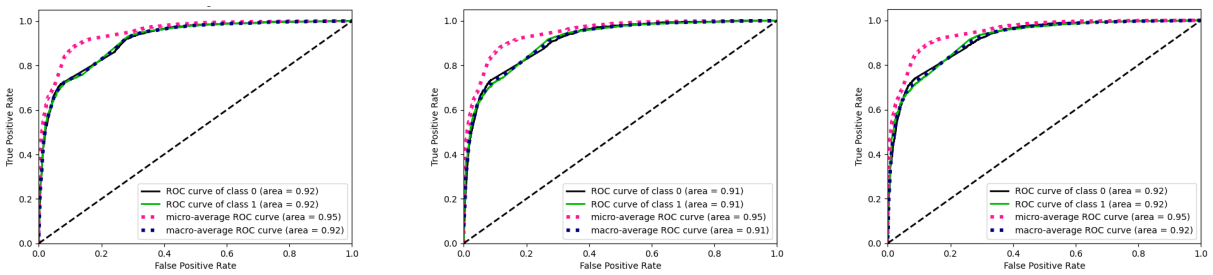
(a) GloVe (b) FastText (c) Word2Vec

Figure 34: ROC curves for AMI dataset



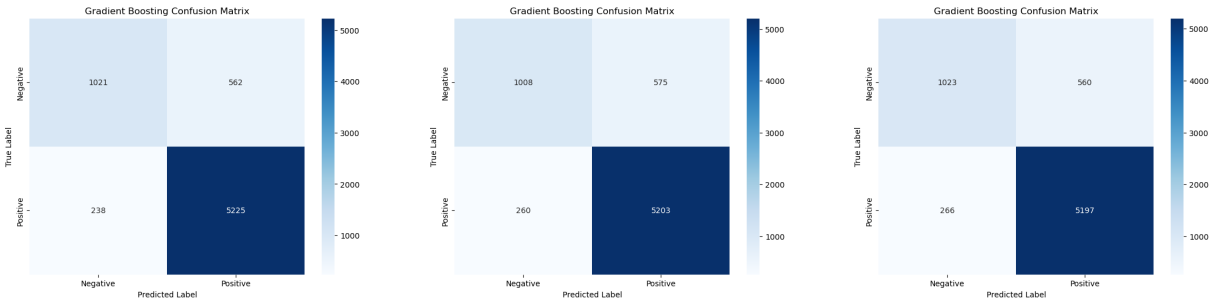
(a) GloVe (b) FastText (c) Word2Vec

Figure 35: Confusion Matrix for AMI dataset


True Positive Rate
False Positive Rate
ROC curve of class 0 (area = 0.92)
ROC curve of class 1 (area = 0.92)
micro-average ROC curve (area = 0.95)
macro-average ROC curve (area = 0.92)
ROC curve of class 0 (area = 0.91)
ROC curve of class 1 (area = 0.91)
micro-average ROC curve (area = 0.95)
macro-average ROC curve (area = 0.91)
ROC curve of class 0 (area = 0.92)
ROC curve of class 1 (area = 0.92)
micro-average ROC curve (area = 0.95)
macro-average ROC curve (area = 0.92)

(a) GloVe (b) FastText (c) Word2Vec

Figure 36: ROC curves for AWC dataset



(a) GloVe (b) FastText (c) Word2Vec

Figure 37: Confusion Matrix for AWC dataset

4 Conclusion and Future works

This thesis has explored and advanced sentiment analysis (SA) techniques for e-commerce platforms, focusing on improving the accuracy and efficiency of customer feedback analysis. By developing and evaluating a range of machine learning (ML), deep learning (DL), and transformer-based models, we have demonstrated the potential of these approaches to extract meaningful insights from Amazon product reviews. Our research objectives were successfully addressed through the development of a comprehensive SA framework, the classification of user opinions into sentiment categories, the investigation of DL approaches, and the fine-tuning of the XLNet model for Amazon review sentiment analysis.

The proposed Att+2D CNN model with an attention mechanism, combined with pretrained word embeddings (GloVe and Word2Vec), achieved superior performance in sentiment classification tasks, outperforming traditional DL models such as 1D CNN, LSTM, BiLSTM, and GRU. This highlights the importance of capturing multi-dimensional patterns in text data and leveraging attention mechanisms to retain critical information in large datasets. Additionally, our hyperparameter optimization experiments using grid search demonstrated the significant impact of tuning on the performance of ML models, with optimized models consistently achieving higher accuracy, precision, recall, and F1-scores.

Among transformer-based models, XLNet emerged as the top performer, achieving the highest accuracy (95.2% on the Amazon Earphones dataset and 95.0% on the Amazon Products dataset) and demonstrating its ability to capture nuanced contextual information through permutation language modeling. However, its computational cost remains a challenge, making DistilBERT a viable alternative for resource-constrained environments due to its competitive performance and faster training times. Our experiments also revealed the importance of batch size and optimizer selection, with a batch size of 16 and the Adam optimizer yielding the best results for XLNet.

The findings of this research have practical implications for e-commerce businesses, enabling them to leverage customer feedback for strategic decision-making and improved customer satisfaction. By combining traditional ML techniques with advanced DL and transformer-based models, this thesis provides a comprehensive toolkit for analyzing customer sentiment at scale.

While this research has made significant contributions to the field of sentiment analysis, several avenues for future work remain. First, optimizing high-performing models

like XLNet for real-time applications by reducing computational costs without compromising accuracy is a critical area of exploration. Techniques such as model distillation, quantization, and pruning could be investigated to achieve this goal.

Second, future research could address challenges such as handling multilingual reviews, managing code-switching, and improving the model's ability to process short and ambiguous reviews. Incorporating explainability techniques into these models would also enhance their interpretability, making them more usable for e-commerce platforms.

Third, expanding the datasets to include a more diverse range of reviews from different product categories and customer demographics would ensure that the models generalize well across various e-commerce platforms. Additionally, exploring the integration of multimodal data, such as combining text with images or user behavior data, could further enhance the robustness of sentiment analysis models.

Finally, future work could evaluate the performance of other word embeddings, transformer-based models, and hybrid architectures to identify additional improvements in sentiment classification. By addressing these challenges and opportunities, the field of sentiment analysis can continue to evolve, providing even greater value to e-commerce businesses and their customers.

5 Publications

Communicated to International Journal

- Shetty AM, Manjaiah D H and Aljunid MF **A Systematic Literature Review of Sentiment Analysis on Amazon Reviews: Trends, Techniques, and Future Directions** *Expert Systems Wiley Publication (Under Review).*

International Journals

- Shetty AM, Manjaiah DH, Aljunid MF. **Fine-tuning XLNet for Amazon review sentiment analysis: A comparative evaluation of transformer models**. Communicated to *ETRI Journal.*, February 2025. DOI: `10.4218/etrij.2024-0318`

- Shetty AM, Aljunid MF, Manjaiah DH. **Unleashing the power of 2D CNN with attention and pre-trained embeddings for enhanced online review analysis**. *International Journal of Computers and Applications*, 46(1):1-12, December 2023. DOI: `10.1080/1206212X.2023.2283647`

International Conference

- Shetty AM, Manjaiah DH, Aljunid MF. **Sentiment Exploring on Feedback of E-commerce Data Using Machine Learning Algorithms**. *Lecture Notes in Electrical Engineering*, December 2023. In book: *Advances in Computing and Information.* DOI: `10.1007/978-981-99-7622-5_8` Presented in **Eighth International Conference on "Emerging Research in Computing, Information, Communication and Applications" (ERCICA - 2023)**, 24th and 25th February 2023 (Offline Mode). Organized by *Nitte Meenakshi Institute of Technology.*

- Shetty, Amrithkala M., Manjaiah D. H., Mohammed Fadhel Aljunid, and Yogesh Km. **Comparative Analysis of Memory-Based Collaborative Filtering and Deep Learning Models for Resolving Cold Start and Data Sparsity Issues in E-commerce Recommender Systems.** *Proceedings of the 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, July 2024. DOI: DOI:10.1109/AIC61668.2024.10730916.

- Shetty, Amrithkala M., Manjaiah D. H., Mohammed Fadhel Aljunid, and Ahammed M. S. Shaik Afzal. **Hyperparameter Optimization of Machine Learning Models Using Grid Search for Amazon Review Sentiment Analysis.** *Proceedings of the Conference on Hyperparameter Optimization of Machine Learning Models Using Grid Search for Amazon Review Sentiment Analysis*, February 2024. DOI:10.1007/978-981-99-7814-4_36. Presented in **4th International Conference on Data Science and Applications (ICDSA 2023)**, organized in In-person and Online (Hybrid Mode) by Malaviya National Institute of Technology Jaipur, India, and technically sponsored by the Soft Computing Research Society on July 14–15, 2023.

- Shetty, Amrithkala M., Manjaiah D. H. **Enhancing Sentiment Analysis of Amazon Reviews with Deep Hybrid Parallel Fusion of CNN and LSTM using Pretrained Word Embeddings Paper accepted**. Presented in **5th International Conference on Data Science and Applications (ICDSA 2024)**, organized in In-person and Online (Hybrid Mode) by Malaviya National Institute of Technology Jaipur, India, and technically sponsored by the Soft Computing Research Society on 17th to 19th July 2024.

- Shetty AM, Manjaiah DH. **Analyzing sentiments in e-commerce: Techniques, applications and challenges**. *International Journal of Science and Research Archive*, 2024, 12(02), 2307–2320. DOI: 10.30574/ijsra.2024.12.2.0843.

Patents

- Shetty AM, Manjaiah DH, Kishore S, Rekha RN. **An AI-powered review mood board system for product designers.** Patent Application Publication, India, 2025, Application No. 202541060502 A, published on 11 July 2025.

References

[1] Alfredo Daza, Néstor Daniel González Rueda, Mirelly Sonia Aguilar Sánchez, Wilmer Filomeno Robles Espíritu, and María Elena Chauca Quiñones. Sentiment analysis on e-commerce product reviews using machine learning and deep learning algorithms: A bibliometric analysis, systematic literature review, challenges and future works. *International Journal of Information Management Data Insights*, 4(2):100267, 2024.

[2] Federico Neri, Carlo Aliprandi, Federico Capeci, Montse Cuadros, and Tomas By. Sentiment analysis on social media. *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 919–926, 2012.

[3] Basant Agarwal and Namita Mittal. *Machine Learning Approach for Sentiment Analysis*, pages 21–45. Springer International Publishing, Cham, 2016.

[4] Saba Naseem, Toqeer Mahmood, Muhammad Asif, Junaid Rashid, Muhammad Umair, and Mohsin Shah. Survey on sentiment analysis of user reviews. 11 2021.

[5] J Sangeetha and U Kumaran. Sentiment analysis of amazon user reviews using a hybrid approach. *Measurement: Sensors*, 27:100790, 2023.

[6] Marouane Birjali, Mohammed Kasri, and Abderrahim beni hssane. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 05 2021.

[7] Charu Aggarwal. *Machine Learning for Text: An Introduction*, pages 1–16. 03 2018.

[8] S. Behdenna, F. Barigou, and G. Belalem. Document level sentiment analysis: A survey. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 4(13), 3 2018.

[9] Hai Dohaiha, PWC Prasad, Angelika Maag, and Abeer Alsadoon. Deep learning for aspect-based sentiment analysis: A comparative review. *Expert Systems with Applications*, 118, 10 2018.

[10] Doaa Mohey El-Din Mohamed Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338, 2018.

[11] Mohammad Tubishat, Norisma Idris, and Mohammad Abushariah. Explicit aspects extraction in sentiment analysis using optimal rules combination. *Future Generation Computer Systems*, 114:448–480, 2021.

[12] Jing Zhang, Aijia Zhang, Dian Liu, and Yiwen Bian. Customer preferences extraction for air purifiers based on fine-grained sentiment analysis of online reviews. *Knowledge-Based Systems*, 228:107259, 2021.

[13] Murtadha Ahmed, Qun Chen, and Zhanhuai Li. Constructing domain-dependent sentiment dictionary for sentiment analysis. *Neural Computing and Applications*, 32:14719–14732, 2020.

[14] Amrithkala M Shetty, DH Manjaiah, et al. Analyzing sentiments in e-commerce: Techniques, applications and challenges. *International Journal of Science and Research Archive*, 12(2):2307–2320, 2024.

[15] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. pages 188–197, 01 2019.

[16] Harsh Thakkar and Dhiren Patel. Approaches for sentiment analysis on twitter: A state-of-art study. 12 2015.

[17] Suhaila Zainudin. Sentiment analysis techniques in recent works. 07 2015.

[18] H. Sankar and V. Subramaniyaswamy. Investigating sentiment analysis using machine learning approach. pages 87–92, 12 2017.

[19] Anna Jurek-Loughrey, Maurice Mulvenna, and Yaxin Bi. Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4, 12 2015.

[20] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134, 2021.

[21] Itisha Gupta and Nisheeth Joshi. Enhanced twitter sentiment analysis using hybrid approach and by accounting local contextual semantic. *Journal of intelligent systems*, 29(1):1611–1625, 2019.

[22] Olamilekan Shobayo, Swethika Sasikumar, Sandhya Makkar, and Obinna Okoyeigbo. Customer sentiments in product reviews: A comparative study with googlepalm. *Analytics*, 3(2):241–254, 2024.

[23] Aman Ullah, Khairullah Khan, Aurangzeb Khan, and Shoukat Ullah. Understanding quality of products from customers' attitude using advanced machine learning methods. *Comput.*, 12:49, 2023.

[24] Passent ElKafrawy, Abdulrahman Mahgoub, Hesham Atef, Abdulrahman Nasser, Mohamed Yasser, Walaa M Medhat, and M Saeed Darweesh. Sentiment analysis: Amazon electronics reviews using bert and textblob. 2023.

[25] V. R. Welgamage, U. A. C. Senarathne, N. H. A. C. Madhubhashani, T. C. Liyanage, and P. P. G. Dinesh Asanka. Overall and feature level sentiment analysis of amazon product reviews using machine learning techniques and web-based chrome plugin. In *2022 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, volume 5, pages 205–210, 2022.

[26] Rosario Catelli, Serena Pelosi, and Massimo Esposito. Lexicon-based vs. bert-based sentiment analysis: A comparative study in italian. *Electronics*, 11(3):374, 2022.

[27] Partha Mukherjee, Youakim Badr, Shreyesh Doppalapudi, Satish Mahadevan Srinivasan, Raghvinder S. Sangwan, and Rahul Sharma. Effect of negation in sentences on sentiment analysis and polarity detection. *Procedia Computer Science*, 185:370–379, 2021.

[28] Xuechun Li, Xueyao Sun, Zewei Xu, and Yifan Zhou. Explainable sentence-level sentiment analysis for amazon product reviews. In *2021 5th International Conference on Imaging, Signal Processing and Communications (ICISPC)*, pages 88–94, 2021.

[29] Rae Yule Kim. Using online reviews for customer sentiment analysis. *IEEE Engineering Management Review*, 49(4):162–168, 2021.

[30] Prasanna Kumar Rangarajan, Bharathi Mohan Gurusamy, Elakkiya R, Charan M, and Rithani M. Automated sentiment classification of amazon product reviews using lstm and bidirectional lstm. pages 1–6, 10 2023.

[31] Myasar R. Tabany and Meriem Gueffal. Sentiment analysis and fake amazon reviews classification using svm supervised machine learning model. *Journal of Advances in Information Technology*, 2024.

[32] Noura A. Semary, Wesam Ahmed, Khalid Amin, Paweł Pławiak, and Mohamed Hammad. Enhancing machine learning-based sentiment analysis through feature extraction techniques. *Plos one*, 19(2):e0294968, 2024.

[33] Haili Wang. Word2vec and svm fusion for advanced sentiment analysis on amazon reviews. *Highlights in Science, Engineering and Technology*, 85:743–749, 2024.

[34] Arman Sarraf. Utilizing nlp sentiment analysis approach to categorize amazon reviews against an extended testing set. *International Journal of Computer (IJC)*, 50(1):107–116.

[35] Aakash, Shagun Gupta, and Amandeep Noliya. Url-based sentiment analysis of product reviews using lstm and gru. *Procedia Computer Science*, 2024.

[36] Vipin Jain, Priyanshu Choudhary, Sachin Arora, Tanishk Mangal, Ankit Choudhary, and Himanshu Kumar. A comprehensive exploration of stack ensembling techniques for amazon product review sentiment analysis. *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 1–5, 2024.

[37] Ameer Hamza, Kashif Bilal Majeed, Muhammad Rashad, Arfan Jaffar, et al. An integrated approach for amazon electronic products reviews by using sentiment analysis. *Bulletin of Business and Economics (BBE)*, 13(2):142–153, 2024.

[38] TR ARAVINDAN, CN VIGNESHWAR, and SUGANESHWARI Ga. Sentiment classification for amazon fine foods reviews using pyspark. In *Recent Developments in Electronics and Communication Systems: Proceedings of the First International Conference on Recent Developments in Electronics and Communication Systems (RDECS-2022)*, volume 32, page 431. IOS Press, 2023.

[39] Sheetal Kusal, Shruti Patil, Aashna Gupta, Harsh Saple, Devashish Jaiswal, Vaishnavi Deshpande, and Ketan Kotecha. Sentiment analysis of product reviews using deep learning and transformer models: A comparative study. In *International Conference on Artificial Intelligence on Textile and Apparel*, pages 183–204. Springer, 2023.

[40] Mohammed Qorich and Rajae El Ouazzani. Text sentiment classification of amazon reviews using word embeddings and convolutional neural networks. *The Journal of Supercomputing*, pages 1–26, 2023.

[41] UB Mahadevaswamy and P Swathi. Sentiment analysis using bidirectional lstm network. *Procedia Computer Science*, 218:45–56, 2023.

[42] Jayakumar Sadhasivam and Ramesh Babu. Sentiment analysis of amazon products using ensemble machine learning algorithm. *International Journal of Mathematical, Engineering and Management Sciences*, 4:508–520, 04 2019.

[43] Monika Agrawal and Nageswara Rao Moparthi. An efficient multiple-word embedding-based cross-domain feature extraction and aspect sentiment classification. *Measurement: Sensors*, 2023.

[44] Wonjoon Kim, Keonwoo Nam, and Youngdoo Son. Categorizing affective response of customer with novel explainable clustering algorithm: The case study of amazon reviews. *Electronic Commerce Research and Applications*, 58:101250, 2023.

[45] James Mutinda, Waweru Mwangi, and George Okeyo. Sentiment analysis of text reviews using lexicon-enhanced bert embedding (lebert) model with convolutional neural network. *Applied Sciences*, 13(3), 2023.

[46] Xiangsen Zhang, Zhongqiang Wu, Ke Liu, Zengshun Zhao, Jinhao Wang, and Chengqin Wu. Text sentiment classification based on bert embedding and sliced multi-head self-attention bi-gru. *Sensors*, 23(3), 2023.

[47] Nassera Habbat, Houda Anoun, and Larbi Hassouni. Combination of gru and cnn deep learning models for sentiment analysis on french customer reviews using xlnet model. *IEEE Engineering Management Review*, 51(1):41–51, 2023.

[48] P Bhuvaneshwari, A Nagaraja Rao, Y Harold Robinson, and MN Thippeswamy. Sentiment analysis for user reviews using bi-lstm self-attention based cnn model. *Multimedia Tools and Applications*, 81(9):12405–12419, 2022.

[49] Ameni Chamekh, Mariem Mahfoudh, and Germain Forestier. Sentiment analysis based on deep learning in e-commerce. In *International Conference on Knowledge Science, Engineering and Management*, pages 498–507. Springer, 2022.

[50] Naveen Kumar Gondhi, Eishita Sharma, Amal H Alharbi, Rohit Verma, Mohd Asif Shah, et al. Efficient long short-term memory-based sentiment analysis of e-commerce reviews. *Computational intelligence and neuroscience*, 2022, 2022.

[51] Amrithkala M Shetty, DH Manjaiah, and Mohammed Fadhel Aljunid. Fine-tuning xlnet for amazon review sentiment analysis: A comparative evaluation of transformer models. *ETRI Journal*, 1225:6463, 2025.

[52] Peter Atandoh, Zhang Fengli, Adu-Gyamfi Daniel, Atandoh Paul, and Nuhoho Raphael. Integrated deep learning paradigm for document-based sentiment analysis. *Journal of King Saud University - Computer and Information Sciences*, 35:101578, 05 2023.

[53] Saman Iftikhar, Bandar Alluhaybi, Mohammed Suliman, Ammar Saeed, and Kiran Fatima. Amazon products reviews classification based on machine learning, deep learning methods and bert. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 21:1084, 10 2023.

[54] Prasanna Kumar Rangarajan, Bharathi Mohan Gurusamy, Elakkiya R, Charan M, and Rithani M. Automated sentiment classification of amazon product reviews using lstm and bidirectional lstm. pages 1–6, 10 2023.

[55] Mohammed AM Ali, DRSN Lokhande, and Safwan AS Alshaibani. Developing hybrid model for analyzing sentiment of textual data for amazon product reviews using deep learning. *Journal of Harbin Engineering University*, 44(7), 2023.

[56] Razin Sumyta Monsoor, Tania Sultana Tamanna, Salequzzaman Khan, Shehrin Hoque, Mahdi Islam, Md Humaion Kabir Mehedi, and Annajiat Alim Rasel. Sentiment analysis of amazon reviews using machine learning classifier. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE, 2023.

[57] Xiaobing Sun and Wei Lu. Understanding attention for text classification. pages 3418–3428, 01 2020.

[58] Griko Nibras. Amazon cell phones reviews. `https://www.kaggle.com/dsv/861634`, 2019.

[59] Meet Nagadia. Amazon kindle book review for sentiment analysis. `https://www.kaggle.com/datasets/meetnagadia/amazon-kindle-book-review-for-sentiment-analysis`, 2023.

[60] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[61] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[62] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

Research Scholar:

Amrithkala M Shetty,
Research Scholar,
Department of Computer Science,
Mangalore University.

Research Supervisor:

Dr. Manjaiah D.H,
Senior Professor,
Department of Computer Science,
Mangalore University.

Chairman:

Dr. Manjaiah D.H,
Senior Professor,
Department of Computer Science,
Mangalore University.